



# Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1

OASIS Standard, March 8<sup>th</sup> 2005

**Document identifier:**

wsdm-muws-part1-1.0

**Location:**

<http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf>

**Editor:**

*William Vambenepe, Hewlett-Packard* <[vbp@hp.com](mailto:vbp@hp.com)>

**Abstract:**

There are two specifications produced by the Web Services Distributed Management technical committee: Management *Using* Web services (MUWS) and Management *Of* Web services (MOWS, see [MOWS]). This document is part of MUWS.

MUWS defines how an Information Technology resource connected to a network provides manageability interfaces such that the IT resource can be managed locally and from remote locations using Web services technologies.

MUWS is composed of two parts. This document is MUWS part 1 and provides the fundamental concepts for management using Web services. MUWS part 2 [MUWS Part 2] provides specific messaging formats used to enable the interoperability of MUWS implementations. MUWS part 2 depends on MUWS part 1, while part 1 is independent from part 2.

**Status:**

This document is an OASIS standard.

Committee members should send comments on this specification to the [wsdm@lists.oasis-open.org](mailto:wsdm@lists.oasis-open.org) list. Others should subscribe to and send comments to the [wsdm-comment@lists.oasis-open.org](mailto:wsdm-comment@lists.oasis-open.org) list. To subscribe, send an email message to [wsdm-comment-request@lists.oasis-open.org](mailto:wsdm-comment-request@lists.oasis-open.org), with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the WSDM TC web page (<http://www.oasis-open.org/committees/wsdm/>).

The errata document for this specification is maintained at:

<http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0-errata.pdf>

## Table of Contents

38	1	Introduction.....	3
39	1.1	Terminology .....	3
40	1.2	Notational conventions .....	4
41	2	Architecture .....	6
42	2.1	Focus on Resources.....	6
43	2.1.1	Capabilities for Management .....	7
44	2.1.2	Isolation from Implementation.....	7
45	2.2	Composability.....	7
46	2.2.1	Low-end to High-end Manageability.....	9
47	2.3	Formal Representation of the Architecture.....	9
48	3	Usage of the Web Services Platform.....	11
49	3.1	Properties.....	11
50	3.2	Operations .....	11
51	3.3	Events.....	11
52	3.4	Metadata.....	12
53	3.5	Addressing.....	12
54	3.6	Security.....	12
55	4	Common Information Items.....	14
56	4.1	WSDM Event Format.....	14
57	4.1.1	XML Representation of the event.....	14
58	4.2	Manageability Endpoint Reference .....	15
59	5	Capabilities.....	16
60	5.1	Identity .....	16
61	5.1.1	Definition.....	16
62	5.1.2	Properties .....	16
63	5.2	Manageability Characteristics.....	18
64	5.2.1	Definition.....	18
65	5.2.2	Properties .....	18
66	5.3	Correlatable Properties.....	19
67	5.3.1	Definition.....	19
68	5.3.2	Information Markup Declarations .....	19
69	5.3.3	Properties .....	20
70	6	Defining a Manageability Interface .....	23
71	7	References.....	24
72	7.1	Normative.....	24
73	7.2	Non-normative.....	24
74	Appendix A.	Acknowledgements .....	26
75	Appendix B.	Notices.....	27
76	Appendix C.	MUWS Part 1 Schema (Normative).....	28
77	Appendix D.	Properties Boolean Match Schema (Normative) .....	30
78			

---

## 79 1 Introduction

80 **Management Using Web Services (MUWS)** enables management of distributed information  
81 technology (IT) resources using Web services. Many distributed IT resources use different  
82 management interfaces. By leveraging Web service technology, MUWS enables easier and more  
83 efficient management of IT resources. This is accomplished by providing a flexible, common  
84 framework for manageability interfaces that leverage key features of Web services protocols.  
85 Universal management and interoperability across the many and various types of distributed IT  
86 resources can be achieved using MUWS.

87 The types of management capabilities exposed by MUWS are the management capabilities  
88 generally expected in systems that manage distributed IT resources. Examples of manageability  
89 functions that can be performed via MUWS include:

- 90 • monitoring the quality of a service
- 91 • enforcing a service level agreement
- 92 • controlling a task
- 93 • managing a resource lifecycle

94 MUWS is designed to meet the requirements defined in the MUWS Requirements document  
95 [MUWS REQS]. Whenever possible, MUWS leverages existing Web services specifications to  
96 ensure interoperability, adoptability, and extensibility.

97 There is a basic set of manageability capabilities defined in this specification. The only capability  
98 required by MUWS is the *Identity* capability defined in section 5.1.

99 To understand the various topics discussed in this specification, the reader should be familiar with  
100 IT management concepts. In addition, the following assumptions are made:

- 101 • The reader is familiar with the Web Services Architecture [WSA].
- 102 • The reader is familiar with XML [XML 1.0 3<sup>rd</sup> Edition], XML Schema [XML Schema Part 1]  
103 [XML Schema Part 2], and XML Namespace [XNS].
- 104 • The reader is familiar with WSDL [WSDL], SOAP [SOAP] and WS-Addressing [WS-  
105 Addressing].
- 106 • The reader is familiar with WS SOAP Message Security [WSS].

107 The text of this specification, along with Appendix C and Appendix D, is normative with the  
108 following exception: the abstract, examples, UML diagrams and any section explicitly marked as  
109 non-normative.

### 110 1.1 Terminology

111 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",  
112 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be  
113 interpreted as described in RFC 2119 [RFC2119].

114 Furthermore, this specification defines and uses the following terms:

115 **Web service endpoint** – an entity providing a destination for Web service messages. A Web  
116 service endpoint has an address (URI) and is described by the content of a WSDL 1.1 port  
117 element. This definition is consistent with the definition provided in the WS-Addressing  
118 specification [WS-Addressing].

119 **Web service interface** – a group of operations described by the content of a WSDL 1.1 portType  
120 element. These operations can provide access to resource properties and metadata.

121 **Resource** – a logical or physical component of some subject domain, for example, a printer, a  
122 magnetic storage disk, an application server, a CRM application or a car engine.

123 **Manageable resource** – a resource capable of supporting one or more standard manageability  
 124 capabilities.

125 **Capability** – a group of properties, operations, events and metadata, associated with identifiable  
 126 semantics and information and exhibiting specific behaviors.

127 **Manageability** – the ability to manage a resource, or the ability of a resource to be managed.

128 **Manageability capability** – a capability associated with one or more management domains.

129 **Standard manageability capability** – a manageability capability that is defined by this  
 130 specification.

131 **Manageability interface** – the composition of one or more manageability capability interfaces.

132 **Manageability capability interface** – a Web service interface representing one manageability  
 133 capability.

134 **Manageability consumer** – a user of manageability capabilities associated with one or more  
 135 manageable resources.

136 **Manageability endpoint** – a Web service endpoint associated with and providing access to a  
 137 manageable resource.

138 **Management domain** – an area of knowledge relative to providing control over, and information  
 139 about, the behavior, health, lifecycle, etc. of manageable resources.

## 140 1.2 Notational conventions

141 This specification uses an informal syntax to describe the XML grammar of the information used  
 142 in defining the management capability interfaces. This syntax uses the following rules:

143

- 144 § The syntax appears as an XML instance, but data types appear instead of  
 145 values.
- 146 § {any} is a placeholder for elements from some other namespace (like ##other  
 147 in the XML Schema).
- 148 § The Cardinality of an attribute, element, or {any}, is indicated by appending  
 149 characters to the item as follows:
 

150 ?	none, or one
151 *	none, or more
152 +	one, or more
153 no character	exactly one
- 154 § Items contained within the square brackets, [ and ], are treated as a group.
- 155 § Items separated by | and grouped within parentheses, ( and ), indicate  
 156 syntactic alternatives.
- 157 § An ellipsis, or three consecutive periods, ..., are used in XML start elements  
 158 to indicate that attributes from some other namespace are allowed.
- 159 § The XML namespace prefixes, defined in section 4, indicate the namespace  
 160 of an attribute or an element.

161

162 A full XML Schema description of the XML information is available in Appendix C of this  
 163 specification.

164

165 When describing an instance of XML information, and in order to refer to an element or an  
 166 attribute, this specification uses a simplified Xpath-like notation that is formally defined as follows:

167

168 Path = '/'? (['@'? (NCName | QName | '\*')] | ['(' (NCName | QName | '\*') ')'] ['/' Path])?

169

170 where:

171 § *NCName* is an XML non-qualified name as defined by the XML  
172 Schema [XMLS]. In this case, the namespace is assumed to default  
173 to the namespace of this specification.

174 § *QName* is an XML qualified name as defines by the XML Schema  
175 [XMLS].

176 § Symbol \* denotes any name match.

177 § Symbol / denotes a path delimiter. When it appears as the first  
178 element of the path, it denotes the root of the XML document.

179 § Symbol @ denotes a reference to an XML attribute. If absent then an  
180 NCName, QName or \* refer to an XML element.

181 § Symbols ( and ) denote a reference to an XML Schema type.

182

183 For example:

184 /E1/E2/@A1 refers to an attribute, A1, of an element, E2, contained  
185 in element E1, which is a root of the XML document.

186

187 E1/ns1:E2/E3 refers to an element, E3, which is contained in element  
188 E2 which is contained in element E1, anywhere in the  
189 XML document. In this case element E2 belongs to the  
190 namespace mapped to the prefix ns1.

191

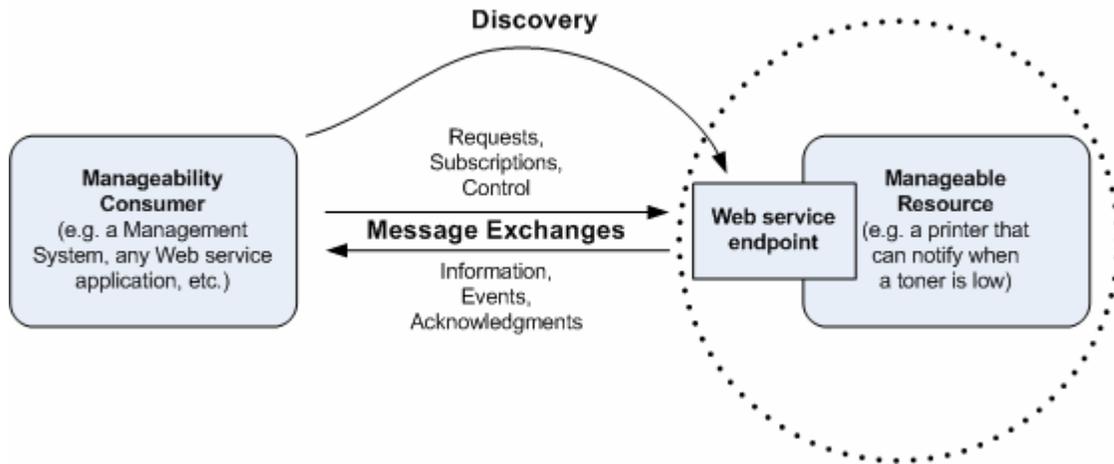
192 (ns2:T1)/E1/ns1:E2/@A1 refers to an attribute, A1, on an element, E2, contained  
193 in element E1, as declared in the XML Schema type  
194 T1. In this case, the target namespace, T1, is mapped  
195 to the prefix ns2.

## 2 Architecture

196

197 This WSDM specification (MUWS) defines how the ability to manage, or, how the *manageability*  
198 *of*, an arbitrary *resource* can be made accessible via *Web services*. In order to achieve this goal,  
199 MUWS is based on a number of Web services specifications, mainly for messaging, description,  
200 discovery, accessing properties, and notifications (section 3). Some of these Web services  
201 specifications are first presented in [MUWS Part 2].

202 The basic concepts of management using Web services can be illustrated by the following figure:



203

204

Figure 1: WSDM Concepts

205 A Web service *endpoint* provides access to a *manageable resource*. An example of a  
206 manageable resource is a printer that indicates when its toner is low, or, a magnetic storage disk  
207 that reports its internal temperature,

208 A *manageability consumer* discovers the Web service endpoint and *exchanges messages* with  
209 the endpoint in order to request information, subscribe to events, or, control the manageable  
210 resource associated with the endpoint. An example of a manageability consumer is a  
211 management system, or, a business automation process, or simply, any Web service application.

212 In order to discover the Web service endpoint providing access to a particular manageable  
213 resource, a manageability consumer first obtains an Endpoint Reference (EPR), as defined by the  
214 WS-Addressing specification [WS-Addressing], and then obtains any other required descriptions,  
215 including, but not limited to, a WSDL document [WSDL], an XML Schema, or a policy document.  
216 MUWS uses the same mechanisms, for obtaining EPRs and their associated descriptions, as  
217 used by regular Web service implementations, and their applications.

218 A Web service endpoint providing access to some manageable resource is called a  
219 *manageability endpoint*.

220 To exchange messages with a manageability endpoint, a manageability consumer needs to  
221 understand all of the required descriptions for the endpoint. The manageability consumer sends  
222 messages targeted to the manageable resource by using information contained in the EPR, for  
223 example, an address and some reference properties (see [WS-Addressing]).

### 2.1 Focus on Resources

224

225 The WSDM specification focuses upon how access is provided to manageable resources.  
226 Essentially, there exists a contract between a manageability consumer and a manageable  
227 resource with respect to the ability of the consumer to understand what messages can be  
228 exchanged between the consumer and the resource. Therefore, the central element and focal

229 point of the WSDM architecture is the manageable resource. The message patterns encapsulate  
230 access to resources into manageable resources instead of exposing message patterns to  
231 indirectly access the resource through agents, proxies, observers, etc.

## 232 2.1.1 Capabilities for Management

233 Manageability is one possible aspect of a resource. For example, a printer can (obviously!) print.  
234 Printing is the functional/operational aspect of the printer. However, the same printer may be able  
235 to indicate if it is on-line, or, if the toner has run out. Such indications compose manageability  
236 capabilities of the printer. A manageable resource may support some number of capabilities.  
237 Each capability has distinct semantics, for example, an ability to describe relationships among  
238 resources or an ability to indicate if the resource is on-line or off-line. An implementation of a  
239 manageable resource provides a set of manageability capabilities via Web service endpoints.

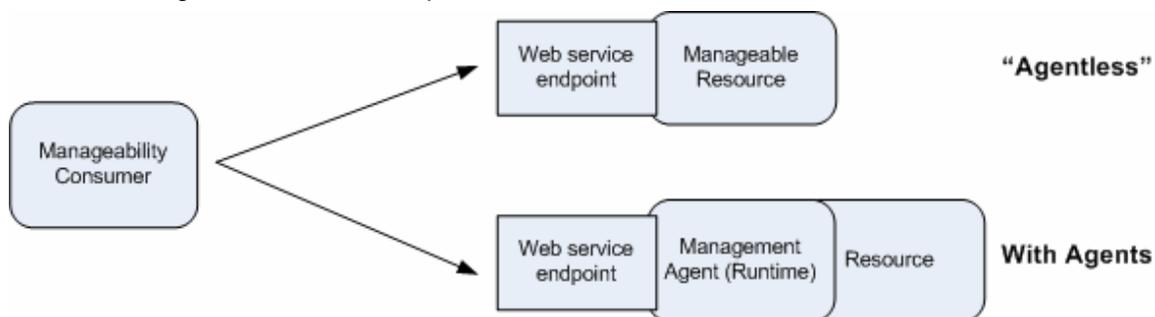
240 In WSDM terms, a *manageability capability*

- 241 § is uniquely identified,
- 242 § has defined semantics (such as those provided by any section in this specification that  
243 describes a new capability),
- 244 § is associated with a set of properties, operations, events (notifications) and metadata  
245 (including policies).

246 Each manageability capability defined in the WSDM specifications is extensible. New capabilities  
247 can be similarly defined, based on a particular resource manageability model, for example, DMTF  
248 CIM. MUWS provides mechanisms, patterns, and refinements, for defining new manageability  
249 capabilities and for discovering, identifying and using capabilities of a manageable resource.

## 250 2.1.2 Isolation from Implementation

251 The WSDM architecture focuses upon the manageable resource. This approach does not restrict  
252 choices of an implementation strategy. Moreover, WSDM isolates the manageability consumer  
253 from implementation specific aspects of a manageable resource or Web service endpoint. For  
254 example, a direct-to-resource, agent-less approach, or, an approach using management agents  
255 are equally valid implementations. Such implementation details are transparent to manageability  
256 consumers. Figure 2 illustrates this point:



257  
258

Figure 2: Isolation from Implementation

## 259 2.2 Composability

260 *Composability* allows a manageable resource's implementation to support a non conflicting mix of  
261 some number of capabilities as well as features provided by the Web services platform. Parts of  
262 the composition incrementally enrich the implementation without incurring disruptions. For  
263 example, a SOAP message sent to a Web service endpoint may result in an order being placed.  
264 A similar SOAP message with WS-Security headers, signed and encrypted, may result in an  
265 order being placed in a secure manner. The mix of the order placement, plus the security  
266 implemented by a Web service endpoint, leveraged message-level composability. In other words,

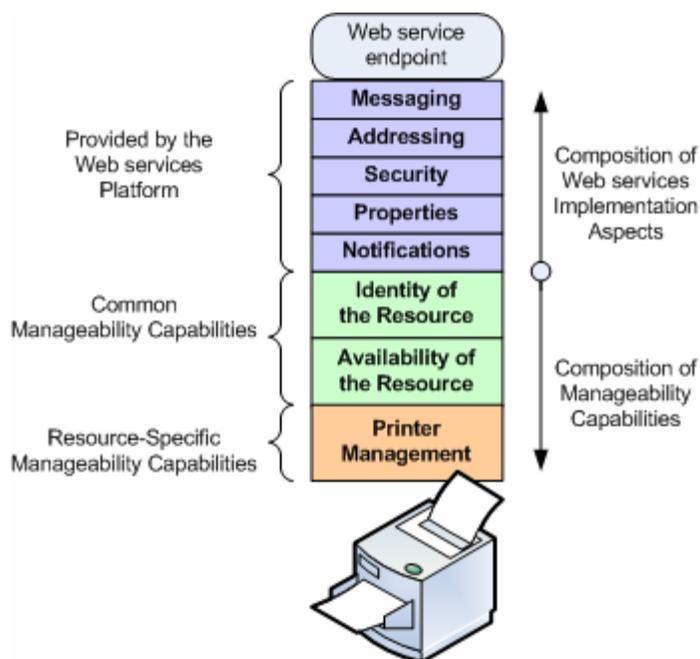
267 the SOAP message is composed of an order placement request, plus the appropriate security  
268 headers, encryption and digital signatures.

269 The implementer of a manageable resource may create an appropriate composition of aspects  
270 and capabilities offered to a manageability consumer via one or more Web service endpoints.  
271 Within the context of WSDM, there are two kinds of composition that can manifest in an  
272 implementation of a manageable resource, as follows:

- 273 1. **Composition of aspects of a Web services implementation** – for example,  
274 messaging, description, discovery, security, asynchronous notifications, etc. These  
275 implementation aspects are provided by the Web services platform and the respective  
276 standards specifications (see section 3).
- 277 2. **Composition of manageability capabilities**, which may be classified into one of two  
278 categories, as follows:
  - 279 a. **Composition of common manageability capabilities** – for example, the ability  
280 to identify manageable resources, the ability to report and notify on a change of  
281 resource availability, or, the ability to report on how resources are related to each  
282 other. Such common manageability capabilities are defined in this specification in  
283 section 4 and in [MUWS Part 2]. Essentially these are base-line enablers of a  
284 richer set of resource manageability. This is similar to how SOAP and HTTP may  
285 be considered baseline enablers of Web services.
  - 286 b. **Composition of resource-specific manageability capabilities** – for example,  
287 an ability to manage printers, or, an ability to manage network-connected  
288 devices. Other specifications define these manageability capabilities based on  
289 the available resource management model, (e.g. DMTF CIM), based on the  
290 needs of the management applications, based on the abilities of the resource  
291 (e.g. WSDM MOWS), or based on the needs of the management application.

292 The whole composition as implemented by a manageable resource is then accessible via a Web  
293 service endpoint. This is illustrated in Figure 3.

294



295

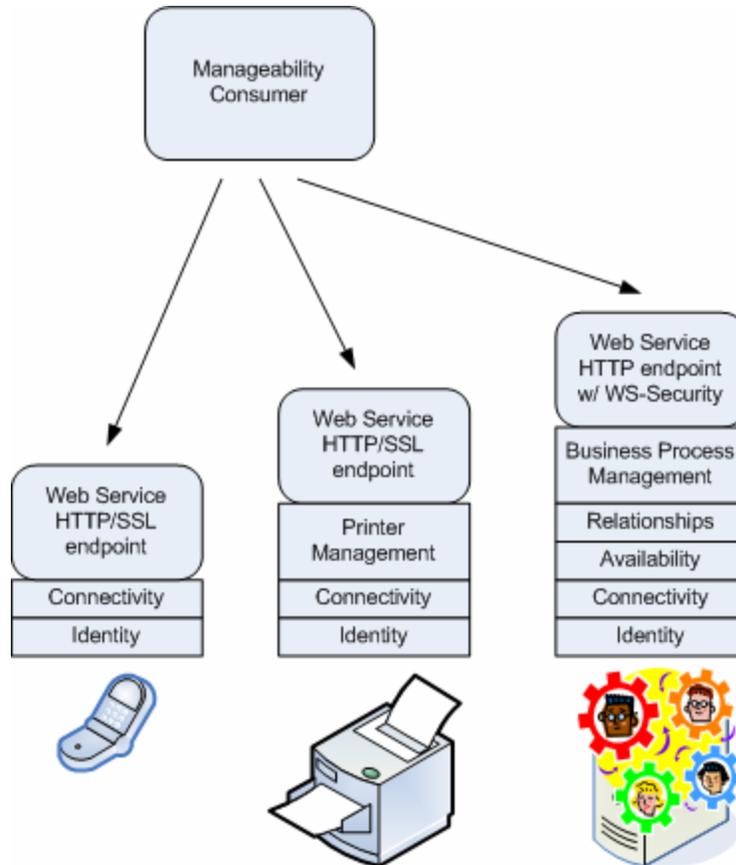
296

297

Figure 3: Composability

298 **2.2.1 Low-end to High-end Manageability**

299 The WSDM architecture provides appropriate coverage from low-end manageability of small  
300 devices like mobile phones, to high-end manageability of very capable components like  
301 application servers and business processes. This range of coverage is achieved by the low  
302 barrier to entry placed upon a WSDM implementation: there are few normative requirements  
303 made by this specification and the specifications it depends on. Also, composability allows for  
304 additional manageability capabilities to be gradually introduced, based upon the availability of  
305 management functions and processing power within an implementation of a manageable  
306 resource. Manageability consumers can discover and make use of composed capabilities as  
307 these capabilities become available. This flexibility is built into the foundation of the WSDM  
308 architecture (Figure 4).

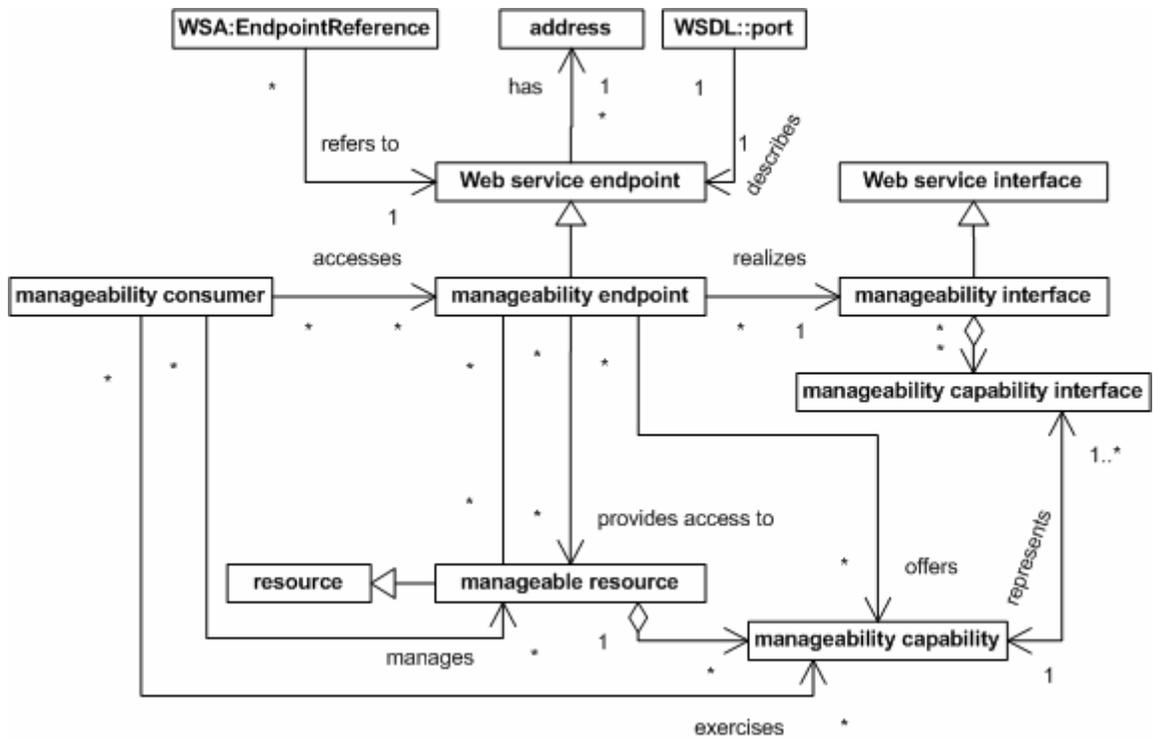


309  
310

Figure 4: Low-end to High-end Manageability

311 **2.3 Formal Representation of the Architecture**

312 The following UML 2.0 model captures the WSDM MUWS concepts within the context of the  
313 WSDL 1.1 [WSDL] and WS-Addressing [WS-Addressing] specifications. Figure 5 provides a  
314 “mind map”, or digest of the concepts described within the WSDM Architecture.



315  
 316  
 317

Figure 5: Formal expression of the WSDM architecture concepts

---

## 318 **3 Usage of the Web Services Platform**

319 As described in section 2, the foundation for MUWS is provided by the Web services platform. A  
320 number of Web services specifications may be composed with the WSDM specifications when  
321 implementing a manageability endpoint for a manageable resource. This and dependent  
322 specifications are used to represent different aspects of a capability: the properties, the  
323 operations, metadata, and events. [MUWS Part 2] introduces additional Web services  
324 specifications to define an interoperable way to represent these capability aspects.

### 325 **3.1 Properties**

326 MUWS uses XML Schema ([XML Schema Part 1], [XML Schema Part 2]) to describe properties.  
327 A MUWS property is represented by a Global Element Declaration (GED). In order to create a  
328 property one MUST provide:

- 329 • the schema for the property,
- 330 • a description (in some form) of the semantics of the property,
- 331 • the cardinality of the property,
- 332 • any relevant metadata for the property.

333 A manageable resource MUST expose an XML document containing, as top-level elements, all  
334 the properties of the manageable resource. This document is called the resource properties  
335 document for the resource.

### 336 **3.2 Operations**

337 MUWS uses [WSDL] to describe operations. The “operations” component of a capability  
338 corresponds to an operation, as defined by WSDL. In order to create an operation one MUST  
339 provide:

- 340 • a WSDL portType containing a WSDL operation corresponding to the operation,
- 341 • a description (in some form) of the semantics of the operation,
- 342 • any relevant metadata for the operation.

### 343 **3.3 Events**

344 Event types (as opposed to instances of event messages) are defined in MUWS by providing the  
345 combination of a “topic” QName and a “message content” Global Element Declaration. The  
346 “topic” QName need not be the QName of the “message content” element. A “topic” or a  
347 “message content” element need not be exclusive to one event. However, the combination of a  
348 “topic” and a “message content” element MUST uniquely identify an event. The “message  
349 content” element represents information that is transmitted as part of a notification message and  
350 corresponds to an event instance. The “topic” provides information about why the event was  
351 generated. In order to create a new event, one MUST provide:

- 352 • the corresponding “topic” and “message content” element,
- 353 • a description (in some form) of the semantics for the “topic” and “message content”  
354 element,
- 355 • any relevant metadata for the event.

356 A manageability endpoint SHOULD offer one or more events that correspond to a change in the  
357 properties it supports.

## 358 **3.4 Metadata**

359 MUWS allows definition of metadata on properties and operations. One such metadata item on  
360 properties is whether it is *Mutable*. Mutability is defined as an indication of whether the value of a  
361 property can change over time. Another metadata item on a property is whether it is *Modifiable*.  
362 Modifiability is defined as an indication of whether the value of a property can be set explicitly, as  
363 opposed to can not be set at all, or, can be set only as a side-effect of setting some other  
364 property. Finally, a *Capability* is a metadata item that can be attached to a property, an operation  
365 or an event. This metadata item contains a unique identifier for the capability. [MUWS Part 2]  
366 describes additional metadata items.

367 For each property introduced in this specification, the value of these metadata items is described.  
368 However, MUWS does not specify if, or how, the value is made available to a consumer.

## 369 **3.5 Addressing**

370 MUWS makes use of the endpoint reference (EPR) construct, as defined in [WS-Addressing]. In  
371 addition, MUWS-compliant messages MUST comply with the rules in [WS-Addressing] regarding  
372 the use of SOAP headers, and, regarding how the content of the EPR constrains the messages  
373 sent to the endpoint.

## 374 **3.6 Security**

375 When evaluating the security requirements for resource management, it is important to delineate  
376 several aspects of Security technology;

- 377 • Identification: Presentation of a claimed identity
- 378 • Authentication: Verification of proof of asserted identity
- 379 • Authorization: The information and mechanisms to allow appropriate authorized requests  
380 to resources and deny unauthorized requests.
- 381 • Message Integrity: The protection of messages in a message exchange from  
382 unauthorized modification.
- 383 • Data Integrity: The protection of data from unauthorized modification.
- 384 • Data confidentiality
- 385 • Trust

386 A complete security model addressing the requirements listed above needs to be provided for  
387 any management deployment. Profiles for different sets of requirements will be needed to ensure  
388 interoperable deployments.

389 An explicit mapping to an authorization model at deployment time should be provided by a  
390 conformant management application.

391 To address security of messages, MUWS relies on generic Web services security mechanisms,  
392 including transport-level security (e.g. HTTP over SSL), OASIS Web Services Security message-  
393 level security [WSS], etc. The composition of appropriate security specifications and this  
394 specification provides a model for securing the messages exchanged during management using  
395 Web services realized by manageability endpoint implementations. The choice of concrete  
396 security mechanisms should be carried out by the implementers of the manageability endpoints  
397 and may conform to some profile.

398 Within an enterprise MUWS can be deployed like any other specification into the existing  
399 enterprise security model. When managing between enterprises, security will need to be  
400 developed in an ad hoc, pair-wise fashion at a messaging level.

401 This specification defines some metadata items for management. Whenever information related  
402 to management metadata is being relied on, it is important to understand the environment in  
403 which the metadata is being asserted. It may be needed to provide some data integrity  
404 mechanisms to protect the information from unauthorized modification. It may also be needed to

405 implement a set of authorization mechanisms to provide a way of identifying under what  
406 conditions information should be shared.

407

## 4 Common Information Items

408

### 4.1 WSDM Event Format

409

The WSDM Event Format defines an XML format to carry management event information. The format defines a set of basic, consistent data elements that allow different types of management event information to be carried in a consistent manner. The WSDM Event Format provides a basis for programmatic processing, correlation, and interpretation of events from different products, platforms, and management technologies.

414

The WSDM Event Format organizes management event data into three basic categories, the event reporter, the event source, and extensible, event-specific, situation data. Each category contains a few common properties, as found in most management events, and allows for extensible, event-specific data. The WSDM Event Format has a flexible and extensible syntax..

418

To be effective, the WSDM Event Format MUST provide the following essential information:

419

- the identification of the resource experiencing an event, called the source,

420

- the identification of the reporter of an event, known as the reporter. In most cases the source reports its own event, thus the identity of the reporter and the source are the same.

421

422

Typically, further information is also needed to describe the semantics of an event.

423

Additionally, an event MUST contain an *EventId* that is unique across event types within the source. An event may contain additional information related to the situation that has occurred or to the context within which it occurred. For example, message text, severity information or related Application Response Measurement (ARM) instrumentation information. It is RECOMMENDED that a container be used to encapsulate additional information that is significant to an event.

429

The base element of the WSDM Event Format is *muws-p1-xs:ManagementEvent*, as presented in the next section.

430

431

#### 4.1.1 XML Representation of the event

432

The following is the XML representation of the WSDM MUWS management event container.

433

```
<muws-p1-xs:ManagementEvent ...
  muws-p1-xs:ReportTime="xs:dateTime" ?>
435
436   <muws-p1-xs:EventId>xs:anyURI</muws-p1-xs:EventId>
437
438   <muws-p1-xs:SourceComponent ...>
439     <muws-p1-xs:ResourceId>xs:anyURI</muws-p1-xs:ResourceId> ?
440     <muws-p1-xs:ComponentAddress>{any}</muws-p1-xs:ComponentAddress> *
441     {any}*
442   </muws-p1-xs:SourceComponent>
443
444   <muws-p1-xs:ReporterComponent ...>
445     <muws-p1-xs:ResourceID>xs:anyURI</muws-p1-xs:ResourceID> ?
446     <muws-p1-xs:ComponentAddress>{any}</muws-p1-xs:ComponentAddress> *
447     {any}*
448   </muws-p1-xs:ReporterComponent> ?
449   {any}*
450 </muws-p1-xs:ManagementEvent>
```

451

Where the clauses are described as follows:

452

**muws-p1-xs:ManagementEvent:** The wrapper element used for management event messages.

453 **muws-p1-xs:ManagementEvent/@muws-p1-xs:ReportTime:** The date and time when the  
454 event was reported. If the value does not include a time zone designation, or use 'Z' for UTC,  
455 then the value MUST be interpreted as having a time zone of UTC. The value of *ReportTime*  
456 MUST provide a granularity as precise as is supported by the generating platform. This attribute  
457 is RECOMMENDED.

458 **muws-p1-xs:ManagementEvent/muws-p1-xs:EventId:** The primary identifier for an event. This  
459 element MUST be unique within the scope provided by the manageability implementation for the  
460 source resource. This element MAY be used as the primary key for the event. This element is  
461 provided for management functions that require events to have an identifier. It is of type URI and  
462 is REQUIRED.

463 **muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent:** The identification of, or  
464 reference to, the source associated with an event. This element is REQUIRED.

465 **muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent/ResourceId:** A specification  
466 of an identifier of a manageable resource associated with an event. This is an OPTIONAL  
467 property. This property is intended as an identifier to be used, for example, in correlation, so that  
468 management consumers can ensure that information contained in the *muws-p1-*  
469 *xs:ManagementEvent* pertains to a given manageable resource. If provided, this element MUST  
470 correspond to the *muws-p1-xs:ResourceId* property (defined in section 5.1.2) for the source  
471 associated with an event.

472 **muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent/muws-p1-**  
473 **xs:ComponentAddress:** Contains the specific elements used to identify the address of a  
474 component. If this element contains more than one child element, each child element represents  
475 an alternate address of the same source. This element is RECOMMENDED to improve  
476 interoperability.

477 **muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent/muws-p1-**  
478 **xs:ComponentAddress/{any}:** XML open content including any XML representation of the  
479 component address. One commonly used address type is a Web service address, such as an  
480 EPR as defined by [WS-Addressing]. In the case where the source is a manageable resource, it  
481 is RECOMMENDED that the *muws-p1-xs:ManageabilityEndpointReference* element, as defined  
482 in section 4.2, be used as the address type.

483 **muws-p1-xs:ManagementEvent/muws-p1-xs:ReporterComponent:** Provides the identification  
484 of, or reference to, the reporter associated with an event. This is a REQUIRED property only if the  
485 reporter is different from the source. Otherwise, this element is OPTIONAL. When this element is  
486 absent the reporter is asserted to be the same as the source. The content of this element is the  
487 same as the content of the *ManagementEvent/SourceComponent* element except that the  
488 definitions apply to the reporter rather than the source.

489 **muws-p1-xs:ManagementEvent/{any}:** XML open content providing a container for additional  
490 data associated with an event. Among other things, this is where the "message content" Global  
491 Element Declaration introduced in section 3.3 is inserted. MUWS Part 2 defines some additional  
492 element that could be included using this wildcard.

## 493 4.2 Manageability Endpoint Reference

494 MUWS defines the following element to represent a reference to a manageability endpoint:

```
495 <muws-p1-xs:ManageabilityEndpointReference>
496   wsa:EndpointReferenceType
497 </muws-p1-xs:ManageabilityEndpointReference>
```

498 The element is an EPR as defined by [WS-Addressing]. The EPR provides a reference to a  
499 manageability endpoint.

---

## 500 5 Capabilities

501 There is a minimum set of manageability capabilities that an implementation of a manageability  
502 endpoint must support in order to comply with the MUWS specification.

503 A manageability capability defines properties, operations and events to support domain-specific  
504 tasks. Details of a manageability capability are exposed by a manageable resource.

505 A manageable resource MAY also define a new resource-specific manageability capability.

506 A manageable resource SHOULD extend a MUWS manageability capability with a resource-  
507 specific manageability capability that uses similar semantics. A manageable resource is not  
508 required to extend a MUWS manageability capability when a resource-specific manageability  
509 capability uses different semantics than the set of MUWS manageability capabilities.

510 In this section the following namespaces are used unless otherwise specified. The table below  
511 lists each prefix and a corresponding namespace URI.

Prefix	Namespace
muws-p1-xs	<a href="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd">http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd</a>
pbm	<a href="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-pbm.xsd">http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-pbm.xsd</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
wsa	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>

### 512 5.1 Identity

513 The manageability capability URI for the *Identity* capability is  
514 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity>

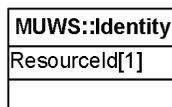
#### 515 5.1.1 Definition

516 The goal of the Identity capability is to establish whether two entities are the same. This is a  
517 required capability and it MUST be provided by every manageability endpoint. Observe that this  
518 requirement does not preclude the manageability endpoint from applying a security policy  
519 preventing some requesters from accessing this, or another, capability.

520 In addition, this capability is used as a “marker” interface enabling a manageability consumer to  
521 learn if an endpoint is a manageability endpoint.

522 Figure 6 shows the UML representation of MUWS Identity.

523



524

525

Figure 6: MUWS Identity

#### 526 5.1.2 Properties

527 The following is the specification of the property defined by the Identity capability.

528 `<muws-p1-xs:ResourceId>xs:anyURI</muws-p1-xs:ResourceId>`

529 The following is an example property instance for the property defined by the Identity capability.

wsdm-muws-part1-1.0

Copyright © OASIS Open 2003-2005. All Rights Reserved.

Page 16 of 30

```
530 <muws-pl-xs:ResourceId>
531   http://example.com/resource/diskDrive/9F34AD35B
532 </muws-pl-xs:ResourceId>
```

533 Note that *ResourceId* is an opaque identifier of a resource managed through a manageability  
534 endpoint. *ResourceId* is a read-only, mandatory property with a cardinality of 1.

535 This property has the following metadata:

536 It is not *Mutable*.

537 It is not *Modifiable*.

538 Its *Capability* is “<http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity>”.

539 The following constraints are applicable to *ResourceId*:

- 540 • Globally unique: A manageability endpoint MUST create the *ResourceId* URI in a way  
541 that ensures that the *ResourceId* is unique to the resource managed through the  
542 manageability endpoint and globally unique. This specification does not prescribe the  
543 means by which global uniqueness is achieved.
- 544 • Uniqueness in time: A *ResourceId* MUST NOT be reused by the implementation of a  
545 manageability endpoint for another resource, even after the original resource no longer  
546 exists.
- 547 • Consistency across endpoints: An implementation of a manageability endpoint SHOULD  
548 use a *ResourceId* that is suggested by the characteristics of a resource. This is possible  
549 when, for example, a *ResourceId* is retrievable from a resource by a manageability  
550 endpoint, or, an application of MUWS to a given domain specifies a method for building a  
551 *ResourceId* based upon characteristics of resources populating the domain. It is not  
552 guaranteed that different manageability endpoints associated with the same resource will,  
553 in all cases, return the same *ResourceId*.
- 554 • Consistency within an endpoint: An implementation that exposes several manageability  
555 endpoints for the same resource MUST report the same *ResourceId* at each  
556 manageability endpoint.
- 557 • Persistence: A manageability endpoint SHOULD return the same *ResourceId* during the  
558 entire lifetime of the manageability endpoint, including across power cycles of the  
559 manageability endpoint. Resources that are not able to persist a *ResourceId* across  
560 power cycles of a manageability endpoint SHOULD try to provide a consistent  
561 *ResourceId* via predictable identifier generation or delegation of identity assignment.
- 562 • Equality: If two reported *ResourceIds* are equal, then the consumer knows that the two  
563 manageability endpoints represent the same resource. The converse proposition is not  
564 necessarily true: two different *ResourceIds* could conceivably correspond to the same  
565 resource. It is strongly RECOMMENDED that this condition be avoided in a conscious  
566 and deliberate manner, as some managers may not be able to distinguish that two  
567 different reported identifiers are, in fact, associated with the same manageable resource.  
568 Thus, manageability consumers would be forced to treat every identifier as corresponding  
569 to a unique manageable resource.

570 Note that a manageability consumer MUST NOT assume that two manageability endpoints  
571 represent two different resources solely because the two reported *ResourceIds* are different.

572 Since the *ResourceId* is defined as opaque, this specification does not allow a consumer to infer  
573 any characteristic of a resource by examining a *ResourceId*, other than comparing the  
574 *ResourceId* to another *ResourceId* as one way of establishing oneness. For example, one  
575 possible way to construct a *ResourceId* and ensure its uniqueness is to use a UUID wrapped in a  
576 URI.

577 Note that this specification does not define equivalence of URIs and the consumer should decide  
578 which level of the comparison ladder, as defined in section 6 of [RFC2396bis], is appropriate to  
579 use for this comparison.

580 MUWS defines an additional mechanism for establishing oneness of two resources. This  
581 mechanism, called *Correlatable Properties* is defined in the section 5.3.

## 582 5.2 Manageability Characteristics

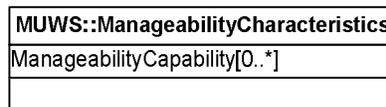
583 The manageability capability URI for the *Manageability Characteristics* capability is  
584 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics>

### 585 5.2.1 Definition

586 The Manageability Characteristics capability defines properties providing information about the  
587 characteristics of a manageability endpoint implementation rather than the resource.

588 Figure 7 shows the UML representation of MUWS Manageability Characteristics.

589



590

591

Figure 7: MUWS Manageability Characteristics

### 592 5.2.2 Properties

593 The following is the specification of the property defined by the Manageability Characteristics  
594 capability.

```
595 <muws-pl-xs:ManageabilityCapability>
596   xs:anyURI
597 </muws-pl-xs:ManageabilityCapability>*
```

598 The following are example of property instances for the property defined by the *Manageability*  
599 *Characteristics* capability.

```
600 <muws-pl-xs:ManageabilityCapability>
601   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity
602 </muws-pl-xs:ManageabilityCapability>
603 <muws-pl-xs:ManageabilityCapability>
604   http://example.com/capabilities/FooCapability
605 </muws-pl-xs:ManageabilityCapability>
```

606 Note that **ManageabilityCapability** contains a URI identifying a manageability capability that is  
607 supported by a manageable resource. The cardinality of this property is zero to unbounded.

608 This property has the following metadata:

609 It is not *Mutable*.

610 It is not *Modifiable*.

611 Its *Capability* is “[http://docs.oasis-  
612 open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics](http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics)”.

613 A manageability interface is said to provide a capability if it supports all of the required properties,  
614 events, operations and metadata defined by the capability. This does not preclude the  
615 manageability endpoint from applying a security policy preventing some requesters from  
616 accessing this, or another, capability.

617 There SHOULD be one *ManageabilityCapability* property instance for each manageability  
618 capability provided by a manageability interface. For capabilities extending a base capability, both  
619 the extension and the base capability MUST be listed. Marking a property, operation or event as  
620 part of a capability is considered a hint for the consumer of a manageability endpoint. The  
621 meaning of such a hint is defined by the capability. As a result, the *ManageabilityCapability*

622 property facilitates discovery and introspection by providing a hint to the manageability consumer  
623 about what requests can be sent to the manageability endpoint.

## 624 5.3 Correlatable Properties

625 The manageability capability URI for the *Correlatable Properties* capability is  
626 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/CorrelatableProperties>

### 627 5.3.1 Definition

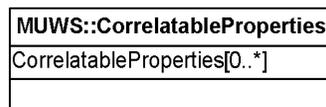
628 The *Correlatable Properties* capability allows a manageability endpoint to expose its  
629 understanding of which property values could be compared when establishing that the  
630 manageability endpoint in question and another manageability endpoint correspond to the same  
631 resource. This is especially useful in the case where the two manageability endpoints are unable  
632 to return the same *ResourceID* for a resource. For example, one manageability endpoint may  
633 enable a temperature control capability for a SCSI hard disk drive, and another manageability  
634 endpoint may enable a capacity management capability for the same SCSI hard disk drive. Each  
635 manageability endpoint may return its own unique *ResourceID* due to implementation  
636 requirements or constraints (e.g. firmware). However, implementers of a manageability endpoint  
637 may be aware of some unique resource-specific property values that can indicate if two  
638 manageability endpoints correspond to the same resource. In the SCSI example, correlatable  
639 properties could be host IP, bus #, channel #, SCSI ID, LUN ID. If the values of those property  
640 instances match, then one could be fairly certain that multiple manageability paths are provided to  
641 the same SCSI resource.

642 Using the *CorrelatableProperties* capability, both manageability endpoints may expose their  
643 understanding of what resource property values need to match in order to establish a correlation  
644 between manageable resources. The manageability consumer uses this information to evaluate  
645 and establish such a correlation.

646 Note that if the *ResourceIDs* returned by both manageability endpoints are the same but the  
647 correlatable properties do not match, then the resources should be considered the same, as the  
648 Identity capability takes precedence over *Correlatable Properties* capability. Typically,  
649 manageability consumers will not evaluate correlatable properties if the two manageability  
650 endpoints return the same *ResourceID*.

651 Figure 8 shows the UML representation of MUWS *Correlatable Properties*.

652



653

654

Figure 8: MUWS *Correlatable Properties*

655 The exposure of the information provided as part of this capability allows clients to understand the  
656 information used to uniquely identify the resource. This may allow a nefarious client to spoof the  
657 presence of the resource. This is particularly true if it is obvious how to generate or construct the  
658 *ResourceID* from these properties. These properties should be used and exposed with this risk in  
659 mind. The *CorrelateableProperties* property should receive the same level of protection as the  
660 *ResourceID*.

### 661 5.3.2 Information Markup Declarations

662 There are three elements, as defined by this specification, providing a simple property boolean  
663 match (PBM) dialect that can be used to express a correlation condition for correlatable  
664 properties. This condition is expressed based on values of properties of the two resources that  
665 are compared through the correlatable properties mechanism. These elements are defined in a  
666 separate namespace, from the rest of the MUWS specification, as follows:

wsdm-muws-part1-1.0

Copyright © OASIS Open 2003-2005. All Rights Reserved.

667 `<pbm:Match>xs:QName</pbm:Match>`

668 This element evaluates to true if the values of the properties for the given QName match for the  
669 two resources.

670 `<pbm:MatchAny>( <pbm:Match/> | <pbm:MatchAll> ) * </pbm:MatchAny>`

671 This element evaluates to true if any of the enclosed *Match* and/or *MatchAll* conditions evaluate  
672 to true.

673 `<pbm:MatchAll>( <pbm:Match/> | </pbm:MatchAny> ) * </pbm:MatchAll>`

674 This element evaluates to true if all of the enclosed *Match* and/or *MatchAny* conditions evaluate  
675 to true.

### 676 5.3.3 Properties

677 The following is a definition of the property defined by the *Correlatable Properties* capability.

```
678 <muws-p1-xs:CorrelatableProperties  
679   Dialect="xs:anyURI"  
680   NegativeAssertionPossible="xs:boolean"?>  
681   {any} *  
682 </muws-p1-xs:CorrelatableProperties>*
```

683 This property indicates, from the perspective of the manageability representation, which property  
684 values, conditions and expressions are used to correlate a manageable resource. The cardinality  
685 of the property is zero to unbounded.

686 This property has the following metadata:

687 It is *Mutable*.

688 It is not *Modifiable*.

689 Its *Capability* is "http://docs.oasis-

690 open.org/wsdm/2004/12/muws/capabilities/CorrelatableProperties".

691 The value of this property is the correlation expression. The format of the correlation expression  
692 is determined by the *Dialect* attribute. This specification defines three possible dialect values. An  
693 additional dialect value can be defined to provide additional functionality. A manageability  
694 representation can offer several instances of the *muws-p1-xs:CorrelatableProperties* property,  
695 using the same, or different, dialects. A manageability consumer may evaluate a *muws-p1-xs:*  
696 *CorrelatableProperties* property in any dialect that it understands. Support for a particular dialect  
697 is optional.

698 The dialects defined by this specification are:

- 699 • Simple Property Boolean Match

700 The URI for this dialect is <http://docs.oasis-open.org/wsdm/2004/12/pbm>.

701 The content of the property is as described in section 5.3.2. If all top-level match  
702 conditions evaluate to true, then a correlation between manageable resources is  
703 established.

- 704 • XPath 1.0

705 The URI for this dialect is <http://www.w3.org/TR/1999/REC-xpath-19991116>.

706 The content of the property is an [XPath 1.0] expression. When retrieved as a property  
707 form a manageable resource, the XPath expression is evaluated on properties of another  
708 manageability resource. If the XPath expression evaluates to a Boolean value of *true*, or  
709 if it evaluates to a non-empty, non-boolean value, without any errors, then a correlation is  
710 established between the manageable resources.

- 711 • XPath 2.0

712 The URI for this dialect is <http://www.w3.org/TR/xpath20/>.

713 The content of the property is an [XPath 2.0] expression. This XPath expression is

714 evaluated on a resource properties document of another manageability representation. If  
715 the XPath expression evaluates to a Boolean value of *true*, or if it evaluates to a non-  
716 empty, non-boolean value, without any errors, then a correlation is established between  
717 the manageable resources.

718 The optional *NegativeAssertionPossible* attributes express whether a negative result from the  
719 evaluation of the correlation expression implies that the resources are necessarily different.  
720 The default value is false.

- 721 • If *NegativeAssertionPossible* is *false*, only a positive match is meaningful to the  
722 consumer. In other words, if the correlation expression evaluates successfully,  
723 according to the evaluation rules defined by the dialect, then a consumer can  
724 consider the resource representations to represent the same resource. If the  
725 correlation expression does not evaluate successfully, then the consumer can not  
726 infer whether the resource representations represent different resources.
- 727 • If *NegativeAssertionPossible* is *true*, a positive match still means that the resources  
728 are the same. But a negative match now means that the resources are guaranteed to  
729 NOT be the same.

### 730 5.3.3.1 Examples of use

731 Consider the following two simplified sets of properties, obtained through two different  
732 manageability endpoints:

733 Properties obtained through manageability endpoint ME1:

```
734 <print:PrinterResourcePropDoc>  
735 ...  
736 <print:PrinterModel>PrintCo SuperJet 5000</print:PrinterModel>  
737 <print:Location>Building 42 lower pillar D4</print:Location>  
738 <print:Owner>Sir Printalot</print:Owner>  
739 <print:IPAddress>15.244.62.41</print:IPAddress>  
740 <foo:Name>Baby got ink</foo:Name>  
741 <muws-pl-xs:CorrelatableProperties  
742   Dialect="http://docs.oasis-open.org/wsdm/2004/12/pbm">  
743   <pbm:MatchAny>  
744     <pbm:Match>print:IPAddress</pbm:Match>  
745     <pbm:MatchAll>  
746       <pbm:Match>foo:Name</pbm:Match>  
747       <pbm:Match>print:PrinterModel</pbm:Match>  
748       <pbm:Match>print:Location</pbm:Match>  
749       <pbm:Match>print:Owner</pbm:Match>  
750     </pbm:MatchAll>  
751   </pbm:MatchAny>  
752 </muw-pl-xs:CorreletableProperties>  
753 </print:PrinterResourcePropDoc>
```

754 Properties obtained through manageability endpoint ME2:

```
755 <print:PrinterResourcePropDoc>  
756 ...  
757 <print:PrinterModel>PrintCo UltraJet 40</print:PrinterModel>  
758 <print:Location>Building 42 lower pillar D4</print:Location>  
759 <print:Owner>Sir Printalot</print:Owner>  
760 <print:IPAddress>15.244.10.89</print:IPAddress>  
761 <foo:Name>Baby got ink</foo:Name>  
762 </print:PrinterResourcePropDoc>
```

763 The *CorrelatableProperties* property, as provided through manageability endpoint ME1, asserts  
764 that if a manageability representation provides a view of a resource which either has the same  
765 *IPAddress* as ME1, or, has the same *Name*, *PrinterModel*, *Location*, and *Owner* as ME1, then  
766 these two manageability endpoints represent are the same printer. In this example, since the

767 *IPAddress* doesn't match and the *PrinterModel* is different, the correlation is not established and  
768 the consumer cannot deduce that the two printers are the same.

769 Note that since the *NegativeAssertionPossible* attribute is not specified on *CorrelatableProperties*  
770 it takes the default value of *false*. Therefore, the consumer cannot assume that the resources are  
771 indeed two different printers. At this point, the consumer still cannot infer whether the two  
772 manageability endpoints correspond to the same printer or not.

773 Properties obtained through manageability endpoint ME3:

```
774 <print:PrinterResourcePropDoc>  
775 ...  
776 <muws-p1-xs:CorrelatableProperties  
777   Dialect=http://www.w3.org/TR/1999/REC-xpath-19991116  
778   NegativeAssertionPossible="false">  
779 boolean(/print:PrinterResourcePropDoc/print:LastJob/print:JobID="5622654845  
780 1262") and  
781 boolean(/print:PrinterResourcePropDoc/print:LastJob/print:JobOriginator="15  
782 .244.30.30")  
783 </muw-p1-xs:CorrelatableProperties>  
784 </print:PrinterResourcePropDoc>
```

785 Properties obtained through manageability endpoint ME4:

```
786 <print:PrinterResourcePropDoc>  
787 ...  
788 <print:LastJob>  
789   <print:JobID>56226548451262</print:JobID>  
790   <print:JobOriginator>15.244.30.30</print:JobOriginator>  
791   <print:JobDate>2004-03-11T11:30:56Z</print:JobDate>  
792 </print:LastJob>  
793 </print:PrinterResourcePropDoc>
```

794 The *CorrelatableProperties* property, as provided through manageability endpoint ME3, asserts  
795 that if a manageability endpoint provides a view of a resource for which the *JobID* of the last job is  
796 56226548451262, and the *JobOriginator* of the last job is 15.244.30.30, then these manageability  
797 endpoints represent the same printer. In this example, the condition is satisfied, so the consumer  
798 knows that ME3 and ME4 correspond to the same physical printer. Note that, as the example  
799 shows, with this dialect the consumer only needs to retrieve the *CorrelatableProperties* property  
800 and no other property from ME3 to check correlation. From ME4 it needs to retrieve the  
801 properties needed to evaluate the XPath expression. In this example, *NegativeAssertionPossible*  
802 is set to *false*, thus a negative result would not have guaranteed that the printers behind ME3 and  
803 ME4 are indeed different.

---

## 804 **6 Defining a Manageability Interface**

805 Implementers of manageability endpoints are free to expose additional manageability capabilities  
806 beyond those defined in MUWS. An additional capability is represented by a set of manageability  
807 capability interfaces. The properties defined in a new capability must be defined as XML Schema  
808 Global Element Declarations. The operations defined in a new capability are represented as  
809 WSDL 1.1 operations. Furthermore, a manageability endpoint offering a new capability is free to  
810 ignore all standard manageability capabilities defined by MUWS except for the *Identity* capability.  
811 The MUWS *Identity* capability is REQUIRED.

812 MUWS-compliant manageability endpoints SHOULD also comply with the WS-I Basic Profile  
813 version 1.1 [BP].

---

## 814 7 References

### 815 7.1 Normative

#### 816 [XML1.0 3<sup>rd</sup> Edition]

817 Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,  
818 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>

#### 820 [XML Schema Part 1]

821 Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C  
822 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>

#### 824 [XML Schema Part 2]

825 Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C  
826 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>

#### 828 [XNS]

829 Tim Bray, et al., *Extensible Namespaces in XML*, W3C  
830 Recommendation, January 1999, [http://www.w3.org/TR/REC-xml-](http://www.w3.org/TR/REC-xml-names/)  
831 [names/](http://www.w3.org/TR/REC-xml-names/)

#### 832 [WSDL]

833 Erik Christensen, et al., *Web services Description Language (WSDL)*  
834 *1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wsdl>

835 [WS-Addressing] Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C  
836 Member Submission, August 2004,  
837 <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>

#### 839 [RFC2119]

840 S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
841 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

#### 842 [RFC2396bis]

843 T. Berners-Lee, et al., *Uniform Resource Identifier (URI): Generic*  
844 *Syntax*, IETF RFC 2396bis-04, February 2004,  
<http://www.ietf.org/internet-drafts/draft-fielding-uri-rfc2396bis-04.txt>

### 845 7.2 Non-normative

#### 846 [MOWS]

847 Igor Sedukhin, *Web Services Distributed Management: Management of*  
848 *Web Services (WSDM-MOWS) 1.0*, OASIS Committee Draft, December  
849 2004, <http://docs.oasis-open.org/wsdm/2004/12/wsdm-mows-1.0.pdf>

#### 850 [MUWS Part 2]

851 William Vambenepe, *Web Services Distributed Management:*  
852 *Management using Web Services (MUWS 1.0) Part 2*, OASIS Committee  
853 Draft, December 2004, [http://docs.oasis-open.org/wsdm/2004/12/wsdm-](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf)  
854 [muws-part2-1.0.pdf](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf)

#### 855 [MUWS REQS]

856 Pankaj Kumar, et al., *Requirements – Management Using Web Services*,  
857 Committee Draft, October 2003, [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/6185/WSDM-MUWS-Req-committee-draft-1.0-20031002.pdf)  
858 [open.org/apps/org/workgroup/wsdm/download.php/6185/WSDM-MUWS-](http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/6185/WSDM-MUWS-Req-committee-draft-1.0-20031002.pdf)  
[Req-committee-draft-1.0-20031002.pdf](http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/6185/WSDM-MUWS-Req-committee-draft-1.0-20031002.pdf)

859		
860	<b>[SOAP]</b>	Don Box, et al., <i>Simple Object Access Protocol (SOAP) 1.1</i> , W3C Note, May 2000, <a href="http://www.w3.org/TR/2000/NOTE-SOAP-20000508/">http://www.w3.org/TR/2000/NOTE-SOAP-20000508/</a>
861		
862		
863	<b>[WSA]</b>	David Booth, et al. <i>Web Services Architecture</i> , W3C Working Group Note, February 2004, <a href="http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/">http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/</a>
864		
865		
866		
867	<b>[WSS]</b>	Anthony Nadalin, et al. <i>Web Services Security: SOAP Message Security 1.0</i> , OASIS Standard, March 2004, <a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf</a>
868		
869		
870		
871	<b>[BP]</b>	Keith Ballinger, et al. <i>Basic Profile Version 1.1</i> , WS-I Final Material, August 2004, <a href="http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html">http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html</a>
872		
873		

---

874 **Appendix A. Acknowledgements**

875 The following individuals were members of the committee when the specification was approved  
876 by the technical committee

877 Guru Bhat, Jeff Bohren, Winston Bumpus, Nick Butler, Brian Carroll, Fred Carter, Michael  
878 Clements, David Cox, John DeCarlo, Andreas Dharmawan, Mark Ellison, John Fuller, Paul  
879 Lipton, Heather Kreger, Hal Lockhart, Frederico Maciel, Tom Maguire, Bryan Murray, Richard  
880 Nikula, Mark Peel, Richard Pelavin, Homayoun Pourheidari, Warren Roberts, Karl Schopmeyer,  
881 Igor Sedukhin, David Snelling, Thomas Studwell, William Vambenepe, Andrea Westerinen, Jim  
882 Willits, Zhili Zhang.

883 In addition, the following non-member employees of member companies made contribution to the  
884 specification: Maryann Hondo, Ian Springer.

---

885 **Appendix B. Notices**

886 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
887 that might be claimed to pertain to the implementation or use of the technology described in this  
888 document or the extent to which any license under such rights might or might not be available;  
889 neither does it represent that it has made any effort to identify any such rights. Information on  
890 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
891 website. Copies of claims of rights made available for publication and any assurances of licenses  
892 to be made available, or the result of an attempt made to obtain a general license or permission  
893 for the use of such proprietary rights by implementers or users of this specification, can be  
894 obtained from the OASIS Executive Director.

895 OASIS invites any interested party to bring to its attention any copyrights, patents or patent  
896 applications, or other proprietary rights which may cover technology that may be required to  
897 implement this specification. Please address the information to the OASIS Executive Director.

898 Copyright © OASIS Open 2003-2005. *All Rights Reserved.*

899 This document and translations of it may be copied and furnished to others, and derivative works  
900 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
901 published and distributed, in whole or in part, without restriction of any kind, provided that the  
902 above copyright notice and this paragraph are included on all such copies and derivative works.  
903 However, this document itself does not be modified in any way, such as by removing the  
904 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS  
905 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
906 Property Rights document must be followed, or as required to translate it into languages other  
907 than English.

908 The limited permissions granted above are perpetual and will not be revoked by OASIS or its  
909 successors or assigns.

910 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
911 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO  
912 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE  
913 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
914 PARTICULAR PURPOSE.

---

## Appendix C. MUWS Part 1 Schema (Normative)

915

```
916 <?xml version="1.0" encoding="utf-8"?>
917 <xs:schema
918     targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
919     part1.xsd"
920     xmlns:muws-pl-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
921     part1.xsd"
922     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
923     xmlns:xs="http://www.w3.org/2001/XMLSchema"
924     elementFormDefault="qualified" attributeFormDefault="unqualified">
925
926     <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
927
928     schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
929
930     <xs:element name="ResourceId" type="xs:anyURI"/>
931
932     <xs:complexType name="IdentityPropertiesType">
933         <xs:sequence>
934             <xs:element ref="muws-pl-xs:ResourceId"/>
935         </xs:sequence>
936     </xs:complexType>
937
938     <xs:element name="IdentityProperties"
939         type="muws-pl-xs:IdentityPropertiesType"/>
940
941
942     <xs:element name="ManageabilityCapability" type="xs:anyURI"/>
943
944     <xs:complexType name="ManageabilityCharacteristicsPropertiesType">
945         <xs:sequence>
946             <xs:element ref="muws-pl-xs:ManageabilityCapability"
947                 minOccurs="0" maxOccurs="unbounded"/>
948         </xs:sequence>
949     </xs:complexType>
950
951     <xs:element name="ManageabilityCharacteristicsProperties"
952         type="muws-pl-xs:ManageabilityCharacteristicsPropertiesType"/>
953
954
955     <xs:complexType name="CorrelatablePropertiesType">
956         <xs:sequence>
957             <xs:any minOccurs="0" maxOccurs="unbounded"
958                 namespace="##other" processContents="lax"/>
959         </xs:sequence>
960         <xs:attribute name="Dialect" type="xs:anyURI"/>
961         <xs:attribute name="NegativeAssertionPossible" type="xs:boolean"/>
962         <xs:anyAttribute namespace="##other"/>
963     </xs:complexType>
964
965     <xs:element name="CorrelatableProperties"
966         type="muws-pl-xs:CorrelatablePropertiesType"/>
967
968     <xs:complexType name="CorrelatablePropertiesPropertiesType">
969         <xs:sequence>
970             <xs:element ref="muws-pl-xs:CorrelatableProperties"
971                 minOccurs="0" maxOccurs="unbounded"/>
972         </xs:sequence>
973     </xs:complexType>
974
```

```

975 <xs:element name="CorrelatablePropertiesProperties"
976           type="muws-pl-xs:CorrelatablePropertiesPropertiesType" />
977
978
979 <xs:complexType name="ComponentAddressType">
980   <xs:sequence>
981     <xs:any namespace="##any" processContents="lax" />
982   </xs:sequence>
983 </xs:complexType>
984
985 <xs:complexType name="ComponentType">
986   <xs:sequence>
987     <xs:element name="ResourceId" type="xs:anyURI"
988               minOccurs="0" />
989     <xs:element name="ComponentAddress"
990               type="muws-pl-xs:ComponentAddressType"
991               minOccurs="0" maxOccurs="unbounded" />
992     <xs:any minOccurs="0" maxOccurs="unbounded"
993           namespace="##other" processContents="lax" />
994   </xs:sequence>
995   <xs:anyAttribute namespace="##other" />
996 </xs:complexType>
997
998 <xs:complexType name="ManagementEventType">
999   <xs:sequence>
1000     <xs:element name="EventId" type="xs:anyURI" />
1001     <xs:element name="SourceComponent" type="muws-pl-xs:ComponentType" />
1002     <xs:element name="ReporterComponent" type="muws-pl-xs:ComponentType"
1003               minOccurs="0" />
1004     <xs:any minOccurs="0" maxOccurs="unbounded"
1005           namespace="##other" processContents="lax" />
1006   </xs:sequence>
1007   <xs:attribute name="ReportTime" type="xs:dateTime" use="optional" />
1008   <xs:anyAttribute namespace="##other" />
1009 </xs:complexType>
1010
1011 <xs:element name="ManagementEvent"
1012           type="muws-pl-xs:ManagementEventType" />
1013
1014 <xs:element name="ManageabilityEndpointReference"
1015           type="wsa:EndpointReferenceType" />
1016
1017 </xs:schema>
1018

```

---

## Appendix D. Properties Boolean Match Schema (Normative)

1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-pbm.xsd"
  xmlns:pbm="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-pbm.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Match" type="xs:QName"/>
  <xs:complexType name="MatchAllType">
    <xs:choice>
      <xs:element ref="pbm:Match"/>
      <xs:element ref="pbm:MatchAny"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="MatchAnyType">
    <xs:choice>
      <xs:element ref="pbm:Match"/>
      <xs:element ref="pbm:MatchAll"/>
    </xs:choice>
  </xs:complexType>
  <xs:element name="MatchAll" type="pbm:MatchAllType"/>
  <xs:element name="MatchAny" type="pbm:MatchAnyType"/>
</xs:schema>
```