



Web Services Distributed Management: Management Using Web Services (MUWS 1.1) Part 1

OASIS Standard, 01 August 2006

Document identifier:

wsdm-muws1-1.1-spec-os-01

Location:

<http://docs.oasis-open.org/wsdm/wsdm-muws1-1.1-spec-os-01.pdf>

Technical Committee:

OASIS Web Services Distributed Management TC

Chair(s):

Heather Kreger, IBM, <kreger@us.ibm.com>

Editor:

Vaughn Bullard, AmberPoint, Inc. <vbullard@amberpoint.com>

William Vambenepe, Hewlett-Packard <vp@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 was last revised or approved by the membership of OASIS on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at www.oasis-open.org/committees/wsdm/.

39 For information on whether any patents have been disclosed that may be essential to
40 implementing this specification, and any offers of patent licensing terms, please refer to
41 the Intellectual Property Rights section of the Technical Committee web page
42 (www.oasis-open.org/committees/wsdm/ipr.php).
43 The non-normative errata page for this specification is located at [www.oasis-](http://www.oasis-open.org/committees/wsdm)
44 [open.org/committees/wsdm](http://www.oasis-open.org/committees/wsdm).
45

46 **Table of Contents**

47	1	Introduction	4
48	1.1	Terminology.....	4
49	1.2	Notational conventions.....	5
50	2	Architecture.....	7
51	2.1	Focus on WSDM Resources.....	7
52	2.1.1	Capabilities for Management	8
53	2.1.2	Composition of Resources.....	8
54	2.1.3	Isolation from Implementation.....	9
55	2.2	Composability.....	9
56	2.2.1	Low-end to High-end Manageability	10
57	3	Usage of the Web Services Platform.....	12
58	3.1	Properties	12
59	3.2	Operations.....	12
60	3.3	Events	12
61	3.4	Metadata	13
62	3.5	Addressing	13
63	3.6	Security	13
64	4	Common Information Items.....	15
65	4.1	WSDM Event Format	15
66	4.1.1	XML Representation of the event	15
67	4.2	Manageability Endpoint Reference	16
68	5	Capabilities	17
69	5.1	Identity.....	17
70	5.1.1	Definition	17
71	5.1.2	Properties.....	17
72	5.2	Manageability Characteristics	18
73	5.2.1	Definition	19
74	5.2.2	Properties.....	19
75	5.3	Correlatable Properties	19
76	5.3.1	Definition	19
77	5.3.2	Information Markup Declarations.....	20
78	5.3.3	Properties.....	20
79	6	Defining a Manageability Capability.....	24
80	7	References.....	25
81	7.1	Normative	25
82	7.2	Non-normative.....	25
83	Appendix A.	Acknowledgements	27
84	Appendix B.	Notices.....	28
85	Appendix C.	MUWS Part 1 Schema (Normative)	29
86	Appendix D.	Properties Boolean Match Schema (Normative).....	31
87			

88 1 Introduction

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

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

- 99 • monitoring the quality of a service
- 100 • enforcing a service level agreement
- 101 • controlling a task
- 102 • managing a resource lifecycle

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

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

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

- 110 • The reader is familiar with the Web Services Architecture **[WSA]**.
- 111 • The reader is familiar with XML **[XML 1.0 3rd Edition]**, XML Schema **[XML Schema Part**
112 **1] [XML Schema Part 2]**, and XML Namespace **[XNS]**
- 113 • The reader is familiar with WSDL **[WSDL]**, SOAP **[SOAP]** and WS-Addressing **[WS-**
114 **Addressing]**.
- 115 • The reader is familiar with WS SOAP Message Security **[WSS]**.

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

118 1.1 Terminology

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

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

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

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

129 **IT Resource** – a logical or physical component of some subject domain, for example, a printer, a
130 magnetic storage disk, an application server or a CRM application.

131 **WS Resource** – a resource defined as the actual composition of a resource and a web service
 132 from which the resource can be accessed.

133 **WSDM Resource** -- a resource for which the management aspect is projected as a WSRF
 134 resource. Further usage of the term “resource” shall indicate a reference to a “WSDM resource”
 135 unless so noted.

136 **Manageable resource** – a resource capable of supporting one or more standard manageability
 137 capabilities.

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

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

141 **Manageability capability** – a capability associated with one or more management domains.
 142 This capability is considered to be a resource property.

143 **Standard manageability capability** – a manageability capability that is defined by this
 144 specification.

145 **Manageability interface** – A Web service interface that exposes interfaces for one or more
 146 manageability capabilities.

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

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

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

153 1.2 Notational conventions

154 This specification uses an informal syntax to describe the XML grammar of the information used
 155 in defining the management capabilities. This syntax uses the following rules:

156

- 157 ▪ The syntax appears as an XML instance, but data types appear instead of
 158 values.
- 159 ▪ {any} is a placeholder for elements from some other namespace (like ##other
 160 in the XML Schema).
- 161 ▪ The Cardinality of an attribute, element, or {any}, is indicated by appending
 162 characters to the item as follows:

163 ?	none, or one
164 *	none, or more
165 +	one, or more
166 no character	exactly one
- 167 ▪ Items contained within the square brackets, [and], are treated as a group.
- 168 ▪ Items separated by | and grouped within parentheses, (and), indicate
 169 syntactic alternatives.
- 170 ▪ An ellipsis, or three consecutive periods, ..., are used in XML start elements
 171 to indicate that attributes from some other namespace are allowed.
- 172 ▪ The XML namespace prefixes, defined in section 5, indicate the namespace
 173 of an attribute or an element.

174

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

177

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

180

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

182

183 where:

- 184 ▪ *NCName* is an XML non-qualified name as defined by the XML
185 Schema **[XML Schema Part 1]**. In this case, the namespace is
186 assumed to default to the namespace of this specification.
- 187 ▪ *QName* is an XML qualified name as defines by the XML Schema
188 **[XML Schema Part 1]**.
- 189 ▪ Symbol * denotes any name match.
- 190 ▪ Symbol / denotes a path delimiter. When it appears as the first
191 element of the path, it denotes the root of the XML document.
- 192 ▪ Symbol @ denotes a reference to an XML attribute. If absent then an
193 NCName, QName or * refer to an XML element.
- 194 ▪ Symbols (and) denote a reference to an XML Schema type.

195

196 For example:

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

199

200 E1/ns1:E2/E3 refers to an element, E3, which is contained in element
201 E2 which is contained in element E1, anywhere in the
202 XML document. In this case element E2 belongs to the
203 namespace mapped to the prefix ns1.

204

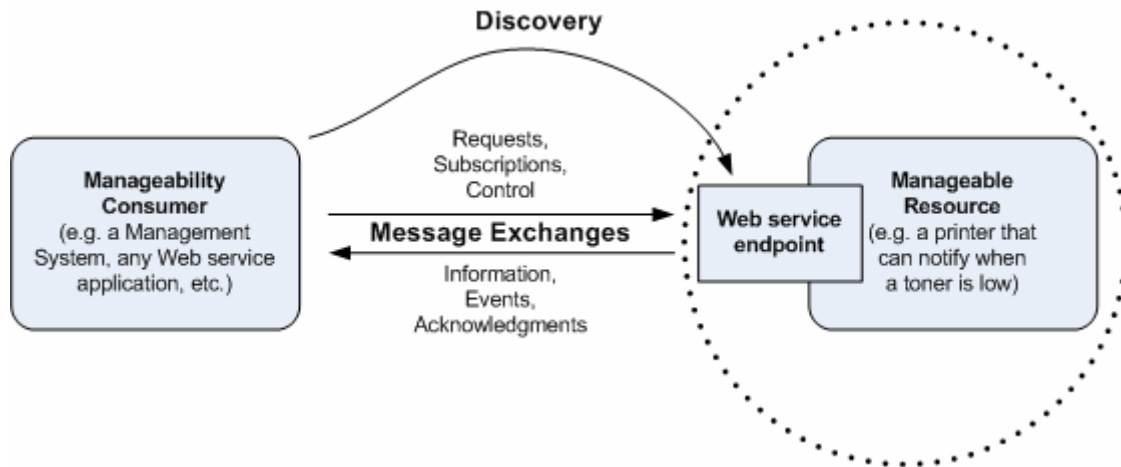
205 (ns2:T1)/E1/ns1:E2/@A1 refers to an attribute, A1, on an element, E2, contained
206 in element E1, as declared in the XML Schema type
207 T1. In this case, the target namespace, T1, is mapped
208 to the prefix ns2.

209

2 Architecture

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

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



216

217

Figure 1: WSDM Concepts

218 A Web service *endpoint* provides access to a *manageable resource*. An example of a
219 manageable resource is a printer that has the capability to alert when its toner is low, or, a
220 magnetic storage disk that reports its internal temperature in the form of a web service operation.

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

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

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

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

2.1 Focus on WSDM Resources

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

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

245 **2.1.1 Capabilities for Management**

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

253 In WSDM terms, a *manageability capability*

- 254 ▪ is uniquely identified in time and environment,
- 255 ▪ has defined semantics (such as those provided by any section in this specification that
256 describes a new capability),
- 257 ▪ is associated with a set of properties, operations, events (notifications) and metadata
258 (including policies).

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

263 **2.1.2 Composition of Resources**

264 As a generic and composable specification, WSDM MUWS can be used whether or not a
265 resource model exists for the resource that is made manageable through MUWS. If a resource
266 model (standard or not) exists for the resource, WSDM MUWS provides ways to expose the
267 elements of this model through Web services standards. In this case, the properties of the
268 manageable resource correspond to the appropriate model elements for this resource, plus the
269 MUWS-defined ResourceId property.

270 In addition, WSDM MUWS Part 2 defines a set of standard model elements, such as elements to
271 represent relationships among resources, a caption, the version, a human-readable description of
272 the resource, the operational status of the resource, etc. These elements can be used if there is
273 no resource model for the resource, in addition to other resource-specific elements that might
274 need to be defined. Even if there is a model for the resource and if the model contains element
275 that semantically overlap the elements defined in MUWS Part 2, the developer might choose to
276 expose the information through both sets of elements in order to maximize interoperability and
277 make the manageability information consumable by more managers.

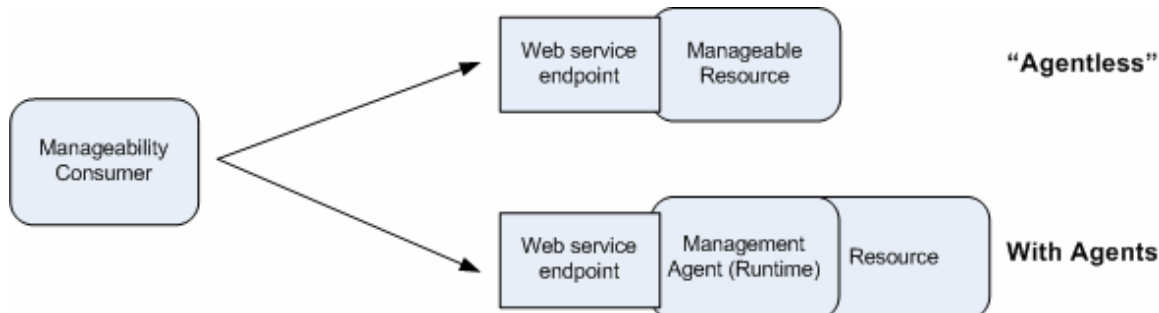
278 In some cases, a resource model only provides a means to represent an individual resource in an
279 XML document. A resource model that is limited in this way does not facilitate the generation of
280 an XML document representing a system comprised of multiple resource instances. For such a
281 case, WS-ServiceGroup provides a means of generating an XML document representing a
282 system of resources. In this case, the system model is exposed by the resource properties
283 document of a WS-ServiceGroup containing the set of resources. Relationships among resources
284 in a WS-ServiceGroup are represented by model elements in a resource model. For example, a
285 relationship can be exposed through a model element defined in a resource model, as a CIM
286 association, or a relationship can be exposed through a MUWS relationship element.

287 Elements of a resource model may be accessed via WSRF operations and received via WS-
288 Notification messages with a level of granularity that is different than the level of granularity used
289 to define a WSDM resource. For example, a single request can be used to retrieve an XML
290 document containing a representation of a system comprised of several WSDM resources.

291 Alternatively, it is possible to use several requests to retrieve a select set of model elements for a
292 WSDM resource.
293

294 2.1.3 Isolation from Implementation

295 The WSDM architecture focuses upon the manageable resource. This approach does not restrict
296 choices of an implementation strategy. Moreover, WSDM isolates the manageability consumer
297 from implementation specific aspects of a manageable resource or Web service endpoint. For
298 example, a direct-to-resource, agent-less approach, or, an approach using management agents
299 are equally valid implementations. Such implementation details are transparent to manageability
300 consumers. **Error! Reference source not found.** illustrates this point:



301
302
303

Figure 2: Isolation from Implementation

304 2.2 Composability

305 *Composability* allows a manageable resource's implementation to support a non conflicting mix of
306 some number of capabilities as well as features provided by the Web services platform. Parts of
307 the composition incrementally enrich the implementation without incurring disruptions. For
308 example, a SOAP message sent to a Web service endpoint may result in an order being placed.
309 A similar SOAP message with WS-Security headers, signed and encrypted, may result in an
310 order being placed in a secure manner. The mix of the order placement, plus the security
311 implemented by a Web service endpoint, leveraged message-level composability. In other words,
312 the SOAP message is composed of an order placement request, plus the appropriate security
313 headers, encryption and digital signatures.

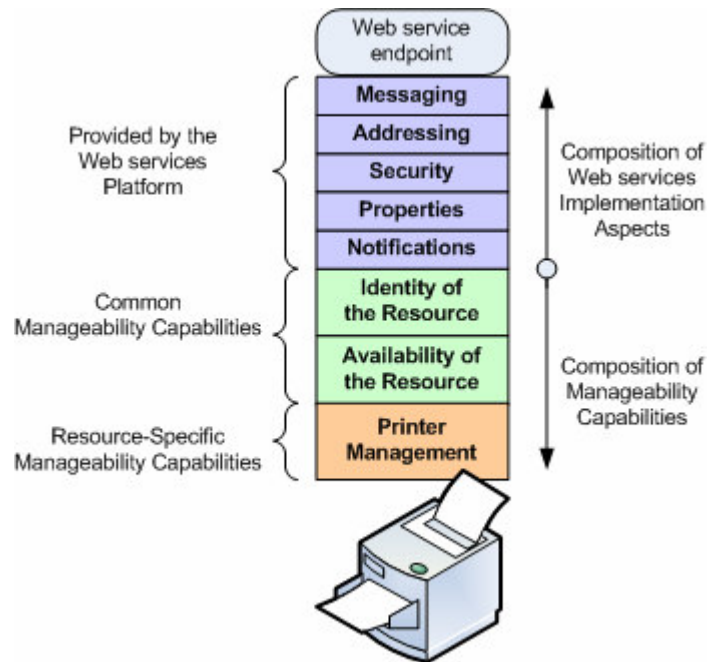
314 The implementer of a manageable resource may create an appropriate composition of aspects
315 and capabilities offered to a manageability consumer via one or more Web service endpoints.
316 Within the context of WSDM, there are two kinds of composition that can manifest in an
317 implementation of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

- 318 1. **Composition of aspects of a Web services implementation** – for example,
319 messaging, description, discovery, security, asynchronous notifications, etc. These
320 implementation aspects are provided by the Web services platform and the respective
321 standards specifications (see section 3).
- 322 2. **Composition of manageability capabilities**, which may be classified into one of two
323 categories, as follows:
 - 324 a. **Composition of common manageability capabilities** – for example, the ability
325 to identify manageable resources, the ability to report and notify on a change of
326 resource availability, or, the ability to report on how resources are related to each
327 other. Such common manageability capabilities are defined in this specification in
328 section 5 and in **[MUWS Part 2]**. Essentially these are base-line enablers of a
329 richer set of resource manageability. This is similar to how SOAP and HTTP may
330 be considered baseline enablers of Web services.

331 b. **Composition of resource-specific manageability capabilities** – for example,
 332 an ability to manage printers, or, an ability to manage network-connected
 333 devices. Other specifications define these manageability capabilities based on
 334 the available resource management model, (e.g. DMTF CIM), based on the
 335 needs of the management applications, based on the abilities of the resource
 336 (e.g. WSDM MOWS), or based on the needs of the management application.

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

339



340

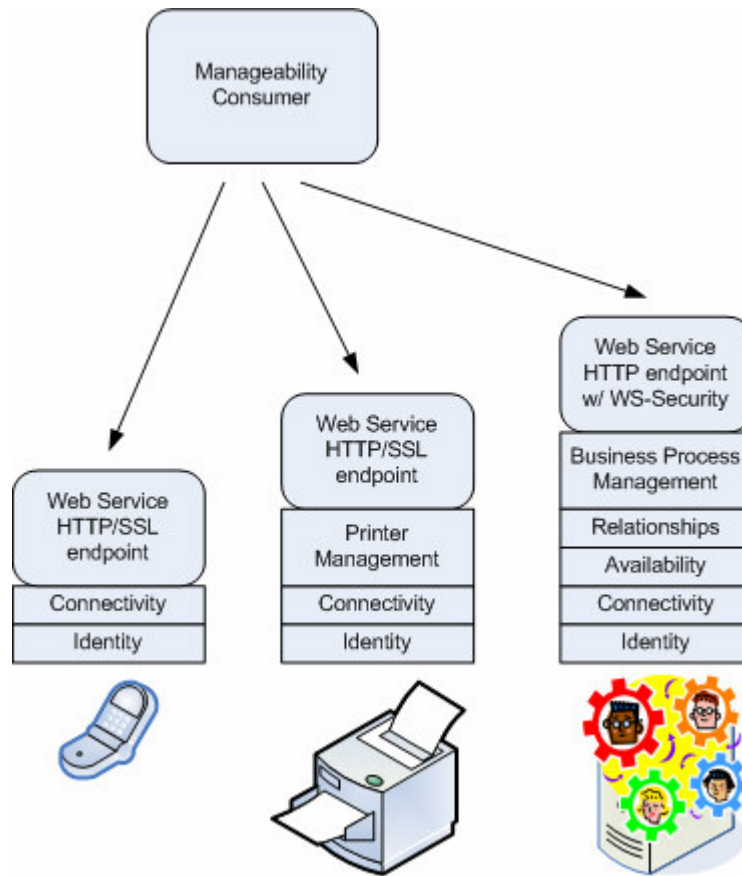
341

Figure 3: Composability

342

343 2.2.1 Low-end to High-end Manageability

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



354
355

Figure 4: Low-end to High-end Manageability

356

3 Usage of the Web Services Platform

357

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

358

359

360

361

362

363

3.1 Properties

364

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

365

366

367

- the schema for the property,

368

- a description (in some form) of the semantics of the property,

369

- the cardinality of the property,

370

- any relevant metadata for the property.

371

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

372

373

374

3.2 Operations

375

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

376

377

378

- a WSDL portType containing a WSDL operation corresponding to the operation,

379

- a description (in some form) of the semantics of the operation,

380

- a WS-Addressing:Action attribute, during the input, output or fault phases that corresponds to a WSA formatted URI,

381

382

- any relevant metadata for the operation.

383

3.3 Events

384

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

385

386

387

388

389

390

391

392

- the corresponding “topic” and “message content” element,

393

- a description (in some form) of the semantics for the “topic” and “message content” element,

394

395

- any relevant metadata for the event.

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

398 **3.4 Metadata**

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

407 For each property introduced in this specification, the value of these metadata items is described.
408 However, MUWS does not specify if, or how, the value is made available to a consumer. A few
409 properties contain actual metadata about a given manageable resource. For example, the
410 ManageabilityCapability capability property as defined in section 5.2.2 **Properties** is one such
411 metadata property.

412 **3.5 Addressing**

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

417 **3.6 Security**

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

- 420 • Identification: Presentation of a claimed identity
- 421 • Authentication: Verification of proof of asserted identity
- 422 • Authorization: The information and mechanisms to allow appropriate authorized requests
423 to resources and deny unauthorized requests.
- 424 • Message Integrity: The protection of messages in a message exchange from
425 unauthorized modification.
- 426 • Data Integrity: The protection of data from unauthorized modification.
- 427 • Data confidentiality
- 428 • Trust

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

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

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

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

444 This specification defines some metadata items for management. Whenever information related
445 to management metadata is being relied on, it is important to understand the environment in
446 which the metadata is being asserted. It may be needed to provide some data integrity
447 mechanisms to protect the information from unauthorized modification. It may also be needed to
448 implement a set of authorization mechanisms to provide a way of identifying under what
449 conditions information should be shared.

450

4 Common Information Items

451

4.1 WSDM Event Format

452

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.

457

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..

461

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

462

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

463

- the identification of the reporter of an event, known as the reporter. In most cases the

464

source reports its own event, thus the identity of the reporter and the source is the same.

465

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

466

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.

471

The base element of the WSDM Event Format is *muws1:ManagementEvent*, as presented in the next section.

473

4.1.1 XML Representation of the event

474

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

475

```
<muws1:ManagementEvent ...
```

476

```
  muws1:ReportTime="xs:dateTime"?)>
```

477

```
  <muws1:EventId>xs:anyURI</muws1:EventId>
```

479

```
  <muws1:SourceComponent ...>
```

481

```
    <muws1:ResourceId>xs:anyURI</muws1:ResourceId> ?
```

482

```
    <muws1:ComponentAddress>{any}</muws1:ComponentAddress> *
```

483

```
    {any}*
```

484

```
  </muws1:SourceComponent>
```

485

```
  <muws1:ReporterComponent ...>
```

487

```
    <muws1:ResourceId>xs:anyURI</muws1:ResourceId> ?
```

488

```
    <muws1:ComponentAddress>{any}</muws1:ComponentAddress> *
```

489

```
    {any}*
```

490

```
  </muws1:ReporterComponent> ?
```

491

```
  {any}*
```

492

```
</muws1:ManagementEvent>
```

493

Where the clauses are described as follows:

494

muws1:ManagementEvent: The wrapper element used for management event messages.

495 **muws1:ManagementEvent/@muws1:ReportTime**: The date and time when the event was
496 reported. If the value does not include a time zone designation, or use 'Z' for UTC, then the value
497 MUST be interpreted as having a time zone of UTC. The value of *ReportTime* MUST provide a
498 granularity as precise as is supported by the generating platform. This attribute is
499 RECOMMENDED.

500 **muws1:ManagementEvent/muws1:EventId**: The primary identifier for an event. This element
501 MUST be unique within the scope provided by the manageability implementation for the source
502 resource. This element MAY be used as the primary key for the event. This element is provided
503 for management functions that require events to have an identifier. It is of type URI and is
504 REQUIRED.

505 **muws1:ManagementEvent/muws1:SourceComponent**: The identification of, or reference to,
506 the source associated with an event. This element is REQUIRED.

507 **muws1:ManagementEvent/muws1:SourceComponent/ResourceId**: A specification of an
508 identifier of a manageable resource associated with an event. This is an OPTIONAL property.
509 This property is intended as an identifier to be used, for example, in correlation, so that
510 management consumers can ensure that information contained in the *muws1:ManagementEvent*
511 pertains to a given manageable resource. If provided, this element MUST correspond to the
512 *muws1:ResourceId* property (defined in section 5.1.2) for the source associated with an event.

513 **muws1:ManagementEvent/muws1:SourceComponent/muws1:ComponentAddress**:
514 Contains the specific elements used to identify the address of a component. If this element
515 contains more than one child element, each child element represents an alternate address of the
516 same source. This element is RECOMMENDED to improve interoperability.

517 **muws1:ManagementEvent/muws1:SourceComponent/muws1:ComponentAddress/{any}**:
518 XML open content including any XML representation of the component address. One commonly
519 used address type is a Web service address, such as an EPR as defined by [WS-Addressing].
520 In the case where the source is a manageable resource, it is RECOMMENDED that the
521 *muws1:ManageabilityEndpointReference* element, as defined in section 4.2, be used as the
522 address type.

523 **muws1:ManagementEvent/muws1:ReporterComponent**: Provides the identification of, or
524 reference to, the reporter associated with an event. This is a REQUIRED property only if the
525 reporter is different from the source. Otherwise, this element is OPTIONAL. When this element is
526 absent the reporter is asserted to be the same as the source. The content of this element is the
527 same as the content of the *ManagementEvent/SourceComponent* element except that the
528 definitions apply to the reporter rather than the source.

529 **muws1:ManagementEvent/{any}**: Provides a container for additional data associated with an
530 event. This is where the "message content" Global Element Declaration introduced in section 3.3
531 is inserted. MUWS Part 2 defines some additional element that could be included using this
532 wildcard.

533 4.2 Manageability Endpoint Reference

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

```
535 <muws1:ManageabilityEndpointReference>  
536   wsa:EndpointReferenceType  
537 </muws1:ManageabilityEndpointReference>
```

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

540 5 Capabilities

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

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

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

546 A manageable resource SHOULD extend a MUWS manageability capability with a resource-
547 specific manageability capability that uses similar semantics. A manageable resource is not
548 required to extend a MUWS manageability capability when a resource-specific manageability
549 capability uses different semantics than the set of MUWS manageability capabilities.

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

Prefix	Namespace
muws1	http://docs.oasis-open.org/wsdm/muws1-2.xsd
pbm	http://docs.oasis-open.org/wsdm/muws/
xs	http://www.w3.org/TR/xmlschema-1/
wsa	http://www.w3.org/2005/08/addressing

552 5.1 Identity

553 The manageability capability URI for the *Identity* capability is
554 <http://docs.oasis-open.org/wsdm/muws/capabilities/Identity>

555 5.1.1 Definition

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

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

562 5.1.2 Properties

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

```
564 <muws1:ResourceId>xs:anyURI</muws1:ResourceId>
```

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

```
566 <muws1:ResourceId>  
567   http://example.com/resource/diskDrive/9F34AD35B  
568 </muws1:ResourceId>
```

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

571 This property has the following metadata:

572 It is not *Mutable*.
573 It is not *Modifiable*.
574 Its *Capability* is “<http://docs.oasis-open.org/wsdm/muws/capabilities/Identity>”.

575 The following constraints are applicable to *ResourceId*:

- 576 • Globally unique: A manageability endpoint MUST create the *ResourceId* URI in a way
577 that ensures that the *ResourceId* is unique to the resource managed through the
578 manageability endpoint and globally unique. This specification does not prescribe the
579 means by which global uniqueness is achieved.
- 580 • Uniqueness in time: A *ResourceId* MUST NOT be reused by the implementation of a
581 manageability endpoint for another resource, even after the original resource no longer
582 exists.
- 583 • Consistency across endpoints: An implementation of a manageability endpoint SHOULD
584 use a *ResourceId* that is suggested by the characteristics of a resource. This is possible
585 when, for example, a *ResourceId* is retrievable from a resource by a manageability
586 endpoint, or, an application of MUWS to a given domain specifies a method for building a
587 *ResourceId* based upon characteristics of resources populating the domain. It is not
588 guaranteed that different manageability endpoints associated with the same resource will,
589 in all cases, return the same *ResourceId*.
- 590 • Consistency within an endpoint: An implementation that exposes several manageability
591 endpoints for the same resource MUST report the same *ResourceId* at each
592 manageability endpoint.
- 593 • Persistence: A manageability endpoint SHOULD return the same *ResourceId* during the
594 entire lifetime of the manageability endpoint, including across power cycles of the
595 manageability endpoint. Resources that are not able to persist a *ResourceId* across
596 power cycles of a manageability endpoint SHOULD try to provide a consistent
597 *ResourceId* via predictable identifier generation or delegation of identity assignment.
- 598 • Equality: If two reported *ResourceIds* are equal, then the consumer knows that the two
599 manageability endpoints represent the same resource. The converse proposition is not
600 necessarily true: two different *ResourceIds* could conceivably correspond to the same
601 resource. It is strongly RECOMMENDED that this condition be avoided in a conscious
602 and deliberate manner, as some managers may not be able to distinguish that two
603 different reported identifiers are, in fact, associated with the same manageable resource.
604 Thus, manageability consumers would be forced to treat every identifier as corresponding
605 to a unique manageable resource.

606 Note that a manageability consumer SHOULD NOT assume that two manageability endpoints
607 represent two different resources solely because the two reported *ResourceIds* are different.

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

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

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

618 5.2 Manageability Characteristics

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

621 5.2.1 Definition

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

624 5.2.2 Properties

625 The following is the specification of the property defined by the Manageability Characteristics
626 capability.

```
627 <muws1:ManageabilityCapability>  
628   xs:anyURI  
629 </muws1:ManageabilityCapability>*
```

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

```
632 <muws1:ManageabilityCapability>  
633   http://docs.oasis-open.org/wsdm/muws/capabilities/Identity  
634 </muws1:ManageabilityCapability>  
635 <muws1:ManageabilityCapability>  
636   http://example.com/capabilities/FooCapability  
637 </muws1:ManageabilityCapability>
```

638 Note that **ManageabilityCapability** contains a URI identifying a manageability capability that is
639 supported by a manageable resource. The ManageabilityCapability property is considered to be a
640 metadata property of the MUWS specification. The cardinality of this property is zero to
641 unbounded.

642 This property has the following metadata:

643 It is not *Mutable*.

644 It is not *Modifiable*.

645 Its *Capability* is "http://docs.oasis-
646 open.org/wsdm/muws/capabilities/ManageabilityCharacteristics".

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

651 There SHOULD be one *ManageabilityCapability* property instance for each manageability
652 capability provided by a manageability interface. For capabilities extending a base capability, both
653 the extension and the base capability MUST be listed. Marking a property, operation or event as
654 part of a capability is considered a hint for the consumer of a manageability endpoint. The
655 meaning of such a hint is defined by the capability. As a result, the *ManageabilityCapability*
656 property facilitates discovery and introspection by providing a hint to the manageability consumer
657 about what requests can be sent to the manageability endpoint.

658 5.3 Correlatable Properties

659 The manageability capability URI for the *Correlatable Properties* capability is
660 http://docs.oasis-open.org/wsdm/muws/capabilities/CorrelatableProperties

661 5.3.1 Definition

662 The *Correlatable Properties* capability allows a manageability endpoint to expose its
663 understanding of which property values could be compared when establishing that the
664 manageability endpoint in question and another manageability endpoint correspond to the same
665 resource. This is especially useful in the case where the two manageability endpoints are unable
666 to return the same *ResourceId* for a resource. For example, one manageability endpoint may

667 enable a temperature control capability for a SCSI hard disk drive, and another manageability
668 endpoint may enable a capacity management capability for the same SCSI hard disk drive. Each
669 manageability endpoint may return its own unique *ResourceID* due to implementation
670 requirements or constraints (e.g. firmware). However, implementers of a manageability endpoint
671 may be aware of some unique resource-specific property values that can indicate if two
672 manageability endpoints correspond to the same resource. In the SCSI example, correlatable
673 properties could be host IP, bus #, channel #, SCSI ID, LUN ID. If the values of those property
674 instances match, then one could be fairly certain that multiple manageability paths are provided to
675 the same SCSI resource. The *CorrelatableProperties* capability is a property that is considered to
676 be metadata.

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

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

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

692 5.3.2 Information Markup Declarations

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

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

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

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

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

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

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

707 5.3.3 Properties

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

```
709 <muws1:CorrelatableProperties  
710   Dialect="xs:anyURI"  
711   NegativeAssertionPossible="xs:boolean"?>  
712   {any} *  
713 </muws1:CorrelatableProperties>*
```

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

717 This property has the following metadata:

718 It is *Mutable*.

719 It is not *Modifiable*.

720 Its *Capability* is “<http://docs.oasis-open.org/wsdm/muws/capabilities/CorrelatableProperties>”.

721 The value of this property is the correlation expression. The format of the correlation expression
722 is determined by the *Dialect* attribute. This specification defines three possible dialect values. An
723 additional dialect value can be defined to provide additional functionality. A manageability
724 representation can offer several instances of the *muws1:CorrelatableProperties* property, using
725 the same, or different, dialects. A manageability consumer may evaluate a *muws1:*
726 *CorrelatableProperties* property in any dialect that it understands. Support for a particular dialect
727 is optional.

728 The dialects defined by this specification are:

- 729 • Simple Property Boolean Match

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

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

- 734 • XPath 1.0

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

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

- 741 • XPath 2.0

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

743 The content of the property is an [XPath 2.0] expression. This XPath expression is
744 evaluated on a resource properties document of another manageability representation. If
745 the XPath expression evaluates to a Boolean value of *true*, or if it evaluates to a non-
746 empty, non-boolean value, without any errors, then a correlation is established between
747 the manageable resources.

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

- 751 • If *NegativeAssertionPossible* is *false*, only a positive match is meaningful to the
752 consumer. In other words, if the correlation expression evaluates successfully,
753 according to the evaluation rules defined by the dialect, then a consumer can
754 consider the resource representations to represent the same resource. If the
755 correlation expression does not evaluate successfully, then the consumer can not
756 infer whether the resource representations represent different resources.
- 757 • If *NegativeAssertionPossible* is *true*, a positive match still means that the resources
758 are the same. But a negative match now means that the resources are guaranteed to
759 NOT be the same.

760 5.3.3.1 Examples of use

761 Consider the following two simplified sets of properties, obtained through two different
762 manageability endpoints:

763 Properties obtained through manageability endpoint ME1:

```

764 <print:PrinterResourcePropDoc>
765   ...
766   <print:PrinterModel>PrintCo SuperJet 5000</print:PrinterModel>
767   <print:Location>Building 42 lower pillar D4</print:Location>
768   <print:Owner>Sir Printalot</print:Owner>
769   <print:IPAddress>15.244.62.41</print:IPAddress>
770   <foo:Name>Baby got ink</foo:Name>
771   <muws1:CorrelatableProperties
772     Dialect="http://docs.oasis-open.org/wsdm/pbm">
773     <pbm:MatchAny>
774       <pbm:Match>print:IPAddress</pbm:Match>
775       <pbm:MatchAll>
776         <pbm:Match>foo:Name</pbm:Match>
777         <pbm:Match>print:PrinterModel</pbm:Match>
778         <pbm:Match>print:Location</pbm:Match>
779         <pbm:Match>print:Owner</pbm:Match>
780       </pbm:MatchAll>
781     </pbm:MatchAny>
782   </muw-pl-xs:CorreletableProperties>
783 </print:PrinterResourcePropDoc>

```

784 Properties obtained through manageability endpoint ME2:

```

785 <print:PrinterResourcePropDoc>
786   ...
787   <print:PrinterModel>PrintCo UltraJet 40</print:PrinterModel>
788   <print:Location>Building 42 lower pillar D4</print:Location>
789   <print:Owner>Sir Printalot</print:Owner>
790   <print:IPAddress>15.244.10.89</print:IPAddress>
791   <foo:Name>Baby got ink</foo:Name>
792 </print:PrinterResourcePropDoc>

```

793 The *CorrelatableProperties* property, as provided through manageability endpoint ME1, asserts
794 that if a manageability representation provides a view of a resource which either has the same
795 *IPAddress* as ME1, or, has the same *Name*, *PrinterModel*, *Location*, and *Owner* as ME1, then
796 these two manageability endpoints represent are the same printer. In this example, since the
797 *IPAddress* doesn't match and the *PrinterModel* is different, the correlation is not established and
798 the consumer cannot deduce that the two printers are the same.

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

803 Properties obtained through manageability endpoint ME3:

```

804 <print:PrinterResourcePropDoc>
805   ...
806   <muws1:CorrelatableProperties
807     Dialect=http://www.w3.org/TR/1999/REC-xpath-19991116
808     NegativeAssertionPossible="false">
809   boolean (/print:PrinterResourcePropDoc/print:LastJob/print:JobID="5622654845
810 1262") and
811   boolean (/print:PrinterResourcePropDoc/print:LastJob/print:JobOriginator="15
812 .244.30.30")
813   </muw-pl-xs:CorrelatableProperties>
814 </print:PrinterResourcePropDoc>

```

815 Properties obtained through manageability endpoint ME4:

```

816 <print:PrinterResourcePropDoc>
817   ...
818   <print:LastJob>

```

```
819 <print:JobID>56226548451262</print:JobID>
820 <print:JobOriginator>15.244.30.30</print:JobOriginator>
821 <print:JobDate>2004-03-11T11:30:56Z</print:JobDate>
822 </print:LastJob>
823 </print:PrinterResourcePropDoc>
```

824 The *CorrelatableProperties* property, as provided through manageability endpoint ME3, asserts
825 that if a manageability endpoint provides a view of a resource for which the *JobID* of the last job is
826 56226548451262, and the *JobOriginator* of the last job is 15.244.30.30, then these manageability
827 endpoints represent the same printer. In this example, the condition is satisfied, so the consumer
828 knows that ME3 and ME4 correspond to the same physical printer. Note that, as the example
829 shows, with this dialect the consumer only needs to retrieve the *CorrelatableProperties* property
830 and no other property from ME3 to check correlation. From ME4 it needs to retrieve the
831 properties needed to evaluate the XPath expression. In this example, *NegativeAssertionPossible*
832 is set to *false*, thus a negative result would not have guaranteed that the printers behind ME3 and
833 ME4 are indeed different.

834 6 Defining a Manageability Capability

835 Implementers of manageability endpoints are free to expose additional manageability capabilities
836 as properties beyond those defined in MUWS. The properties defined in a new capability must
837 be defined as XML Schema Global Element Declarations. The operations defined in a new
838 capability are represented as WSDL 1.1 operations. Furthermore, a manageability endpoint
839 offering a new capability is free to ignore all standard manageability capabilities defined by
840 MUWS except for the *Identity* capability. The MUWS *Identity* capability is REQUIRED.
841 MUWS-compliant manageability endpoints SHOULD also comply with the WS-I Basic Profile
842 version 1.1 **[BP]**.

843 7 References

844 7.1 Normative

- 845 **[XML1.0 3rd Edition]**
846 Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,
847 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>
848
- 849 **[XML Schema Part 1]**
850 Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C
851 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>
852
- 853 **[XML Schema Part 2]**
854 Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C
855 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>
856
- 857 **[XNS]** Tim Bray, et al., *Extensible Namespaces in XML*, W3C
858 Recommendation, January 1999, [http://www.w3.org/TR/REC-xml-](http://www.w3.org/TR/REC-xml-names/)
859 [names/](http://www.w3.org/TR/REC-xml-names/)
860
- 861 **[WSDL]** Erik Christensen, et al., *Web services Description Language (WSDL)*
862 1.1, W3C Note, March 2001, <http://www.w3.org/TR/wsdl>
863
- 864 **[WS-Addressing]** Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C
865 Member Submission, August 2004, <http://www.w3.org/TR/ws-addr-core>
866
- 867 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
868 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
869
- 870 **[RFC2396bis]** T. Berners-Lee, et al., *Uniform Resource Identifier (URI): Generic*
871 *Syntax*, IETF RFC 2396bis-04, February 2004,
872 <http://www.ietf.org/internet-drafts/draft-fielding-uri-rfc2396bis-04.txt>

873 7.2 Non-normative

- 874 **[MOWS]** Kirk Wilson, *Web Services Distributed Management: Management of*
875 *Web Services (WSDM-MOWS) 1.1*, OASIS Committee Draft, February
876 2006, <http://docs.oasis-open.org/wsdm/wsdm-mows-1.1-spec-os-01.pdf>
877
- 878 **[MUWS Part 2]** Vaughn Bullard, *Web Services Distributed Management: Management*
879 *using Web Services (MUWS 1.1) Part 2*, OASIS Committee Draft, March
880 2006, <http://docs.oasis-open.org/wsdm/wsdm-muws2-1.1-spec-os-01.pdf>
881
- 882 **[MUWS REQS]** Pankaj Kumar, et al., *Requirements – Management Using Web Services*,
883 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)
884 [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)
885 [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)
886

887	[SOAP]	Don Box, et al., <i>Simple Object Access Protocol (SOAP) 1.1</i> , W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
888		
889		
890	[WSA]	David Booth, et al. <i>Web Services Architecture</i> , W3C Working Group Note, February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/
891		
892		
893		
894	[WSS]	Anthony Nadalin, et al. <i>Web Services Security: SOAP Message Security 1.0</i> , OASIS Standard, March 2004, http://docs.oasis-
895		
896		open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
897		
898	[BP]	Keith Ballinger, et al. <i>Basic Profile Version 1.1</i> , WS-I Final Material, August 2004, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-
899		
900		24.html

901 **Appendix A. Acknowledgements**

902 WSDM Management Using Web Services Part 1 Version 1.0 Acknowledgements

903 The following individuals were members of the committee when the WSDM MUWS Version 1.0
904 was approved by the technical committee

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

911 In addition, the following non-member employees of member companies made contribution to the
912 specification: Maryann Hondo, Ian Springer, John Gerken, David Melgar, Mitsunori Satomi.

913

914 WSDM Management Using Web Services Part 1 Version 1.1 Acknowledgements

915 The following people made contributions to the WSDM MUWS Version 1.1 specification: Vaughn
916 Bullard, Fred Carter, David Cox, Zulah Eckert, Mark Ellison, Heather Kreger, Frederico Maciel,
917 Bryan Murray, Richard Nikula, Mitsunori Satomi, Thomas Studwell, Kirk Wilson, Zhili Zhang with
918 special thanks to Vaughn Bullard and Mark Ellison as editors.

919 The following individuals were members of the committee while the WSDM MUWS Version 1.1
920 specification was developed and approved by the technical committee: Guru Bhat, Jeff Bohren,
921 Vaughn Bullard, Winston Bumpus, Fred Carter, Michael Clements, David Cox, Zulah Eckert,
922 Mark Ellison, John Fuller, Tony Gullato, Heather Kreger, Richard Landau, Frederico Maciel, Tom
923 Maguire, David Melgar, Bryan Murray, Richard Nikula, Mark Peel, Mitsunori Satomi, Thomas
924 Studwell, William Vambenepe, Kirk Wilson, Zhili Zhang.

925

Appendix B. Notices

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

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

938 Copyright © OASIS Open 2003-2006. *All Rights Reserved.*

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

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

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

Appendix C. MUWS Part 1 Schema (Normative)

955

```
956 <?xml version="1.0" encoding="utf-8"?>
957 <xs:schema xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
958 xmlns:wsa="http://www.w3.org/2005/08/addressing"
959 xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://docs.oasis-
960 open.org/wsdm/muws1-2.xsd" elementFormDefault="qualified"
961 attributeFormDefault="unqualified">
962   <xs:import namespace="http://www.w3.org/2005/08/addressing"
963   schemaLocation="http://www.w3.org/2005/08/addressing/ws-addr.xsd"/>
964   <xs:element name="ResourceId" type="xs:anyURI"/>
965   <xs:element name="ManageabilityCapability" type="xs:anyURI"/>
966   <xs:complexType name="CorrelatablePropertiesType">
967     <xs:sequence>
968       <xs:any namespace="##other" processContents="lax"
969 minOccurs="0" maxOccurs="unbounded"/>
970     </xs:sequence>
971     <xs:attribute name="Dialect" type="xs:anyURI"/>
972     <xs:attribute name="NegativeAssertionPossible" type="xs:boolean"/>
973     <xs:anyAttribute namespace="##other"/>
974   </xs:complexType>
975   <xs:element name="CorrelatableProperties"
976 type="muws1:CorrelatablePropertiesType"/>
977   <xs:complexType name="ComponentAddressType">
978     <xs:sequence>
979       <xs:any namespace="##any" processContents="lax"/>
980     </xs:sequence>
981   </xs:complexType>
982   <xs:complexType name="ComponentType">
983     <xs:sequence>
984       <xs:element name="ResourceId" type="xs:anyURI"
985 minOccurs="0"/>
986       <xs:element name="ComponentAddress"
987 type="muws1:ComponentAddressType" minOccurs="0" maxOccurs="unbounded"/>
988       <xs:any namespace="##other" processContents="lax"
989 minOccurs="0" maxOccurs="unbounded"/>
990     </xs:sequence>
991     <xs:anyAttribute namespace="##other"/>
992   </xs:complexType>
993   <xs:complexType name="ManagementEventType">
994     <xs:sequence>
995       <xs:element name="EventId" type="xs:anyURI"/>
996       <xs:element name="SourceComponent"
997 type="muws1:ComponentType"/>
998       <xs:element name="ReporterComponent"
999 type="muws1:ComponentType" minOccurs="0"/>
1000       <xs:any namespace="##other" processContents="lax"
1001 minOccurs="0" maxOccurs="unbounded"/>
1002     </xs:sequence>
1003     <xs:attribute name="ReportTime" type="xs:dateTime"
1004 use="optional"/>
1005     <xs:anyAttribute namespace="##other"/>
1006   </xs:complexType>
1007   <xs:element name="ManagementEvent" type="muws1:ManagementEventType"/>
1008   <xs:element name="ManageabilityEndpointReference"
1009 type="wsa:EndpointReferenceType"/>
1010   <!-- SCHEMA COPY Material and paste element references below into the
1011 schema of a resource properties document references are provide to insure that
1012 the correct minOccurs/maxOccurs attributes are specified in a resource property
1013 document schema.
1014 : You must import the MUWS Part 1 schema namespace (MUWS1).
```

```
1015
1016      ** Identity Properties      **
1017      <xs:element ref="muws1:ResourceId"/>
1018      ** ManageabilityCharacteristics Properties      **
1019      <xs:element ref="muws1:ManageabilityCapability"
1020                minOccurs="0" maxOccurs="unbounded"/>
1021      ** Correlatable Properties      **
1022      <xs:element ref="muws1:CorrelatableProperties"
1023                minOccurs="0" maxOccurs="unbounded"/>
1024      -->
1025      </xs:schema>
1026
```

Appendix D. Properties Boolean Match Schema (Normative)

1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  targetNamespace="http://docs.oasis-open.org/wsdm/pbm.xsd"
  xmlns:pbm="http://docs.oasis-open.org/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>
```