An Introduction to WSDM

Committee Draft, February 24, 2006

Document identifier:
wsdm-1.0-intro-primer-cd-01

Location:
http://docs.oasis-open.org/wsdm/wsdm-1.0-intro-primer-cd-01.pdf

Editors:
Vaughn Bullard, AmberPoint, Inc. <vbullard@amberpoint.com>
Bryan Murray, Hewlett-Packard <bryan.murray@hp.com>
Kirk Wilson, Computer Associates International <kirk.wilson@ca.com>

Abstract:

This Introduction provides an overview to the WSDM specification and its associated sub-specifications. The introduction is directed towards a wide audience of architects, developers, systems and software integration specialists and users.

In addition, this introduction covers the historical motivations for the creation of WSDM as well as the motivations for why would want to use WSDM as a management specification within their information technology environment.

The introduction provides simple examples of how WSDM can be used in end devices to give the reader ideas of how the WSDM standard can be used in the real world.

This introduction does not provide a definitive source of the WSDM specification. Rather it is intended to provide an easily read and understood summary of the fundamentals of creating and using WSDM-compliant management applications and manageable resources.

Status:

This document is updated periodically on no particular schedule. Send comments to the editor.

Committee members should send comments on this specification to the wsdm@lists.oasis-open.org list. Others should subscribe to and send comments to the wsdm-comment@lists.oasis-open.org list. To subscribe, send an email message to 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 page for this specification is at http://docs.oasis-open.org/wsdm//wsdm-1.0-primer-errata.pdf.

41 # Table of Contents

1 Overview

The WSDM, or *Web Services Distributed Management*, standard is more than a management protocol, SNMP trap handler, or simple distributed management technology. As a standard, it seeks to unify management infrastructures by providing a vendor, platform, network, and protocol neutral framework for enabling management technologies to access and receive notifications of management-enabled resources. Though built upon a standardized suite of XML **[XML]** specifications, it provides features to enable resources that other proprietary management technologies do not. It can be used to standardize management for many devices, from network management devices as well as consumer electronic devices, such as televisions, digital video disc players, and PDAs.

In this introduction, a detailed explanation will be provided of WSDM without providing the 'how' a manufacturer or developer would do the implementation. The explanation goes into the detail of the historical motivation as well as examples usage of WSDM. In section 2.2 , "A Day in the Life of a Web-Enabled Resource" provides real world scenarios of how organizations could possibly implement WSDM as a management framework.

In addition, "The Objectives of WSDM" in section 3, provides a suite of goals the WSDM Standard Technical Committee has set out to achieve. In section 4,"What is WSDM?" a more thorough explanation of the attributes, elements and resources needed are discussed in further detail. In section 5, "The Structure of the WSDM Standard"; the standards management aspect as well as the stack, or suite of technologies, that make up the foundation of WSDM are discussed in detail. Finally, a brief summary is provided as well as an appendix and reference section to links of the base specifications.

2 WSDM Explained

2.1 Motivation for using WSDM

In current Information Technology (IT) environments, there exists a complex collection of heterogeneous systems management technologies and solutions. These heterogeneous IT resources rely upon an ever increasing variety of heterogeneous management technologies. To manage business systems in a more cost effective manner requires the IT operations manager to deal effectively with the complex task of integrating multiple and varied management technologies into a cohesive whole. While providers of management software offer enterprise management solutions that reduce some of this complexity, the offered solutions only provide users with a single point of control. Sometimes, this single point of control exists only within the scope of a specific management domain for a set of tasks or processes. Moreover, it is often difficult to achieve an acceptable level of integration among different enterprise management systems that provides an overall view of cross-application and end-to-end business processes.

Currently, the IT industry and users embrace a variety of standards to integrate management products from multiple vendors. From this situation emerges the need for a standard enabling the integration and accommodation of the full variety of means for managing IT resources. The goal of the WSDM standard is to address this expressed need.

Web services technology was designed to address the general problem of the integration of applications, especially the integration of applications built with a heterogeneous set of implementation technologies and platforms. The mainstream adoption of open, Web services standards created an opportunity for the systems management community to leverage these technologies for the integration of management applications used to manage heterogeneous IT resources. One may choose to apply the same Web services approach used for integrating applications to the problem of integrating management applications and the management of IT resources. By applying the Web services approach, the systems management infrastructure is positioned as a vendor-neutral, platform-independent foundation allowing the use of a common messaging protocol between a manageable resource and a manageability consumer and among manageability consumers themselves. The WSDM standards specify a common messaging protocol for managed resources and their consumers.

The benefits of the WSDM standard are achieved by making the integration of management aspects of diverse IT resources easier and more flexible, and by the coordination of management applications with enterprise business systems. For example, performance metrics of a service are used by a business process when deciding which instance of a Web service to use. By evolving the current management infrastructure to a Web-based standard, the specification enables the migration of existing management processes towards agile management components that easily integrate with business processes. Additional capabilities, for example policy-based management and optimization, are composable into management systems enabling new levels of business adaptivity. WSDM allows management components to become a direct part of a business process. The systems management community can use the same business process technology to develop automated management processes. Integration at this level enables every business to adapt and respond to changing market environments in a more competitive, cost effective and timely manner. Thus, a strategy of utilizing broadly adopted, industrial strength tooling and

runtime technologies will yield a better, more seamless integration among systems management processes, solutions and business applications.

2.2 A Day in the Life of a WSDM-Enabled Resource

To understand a concept, it is often said that one must walk in another's shoes. To this end, this section presents the concept of a day in the life of a resource with WSDM and without it.

The initial vision of WSDM was to standardize the way IT environments interoperate. However, the vision of WSDM as a management framework is widening to a vaster array of possibilities beyond the original intent of WSDM. WSDM seeks to specify that anything within a web services framework could be WSDM management-enabled. That is, being able to collect and manage information from IT resources such as printers and PDAs. In addition, managing consumer electronic devices such as DVD players, televisions, and car radios is more than a possibility. With the rapid adoption of service oriented architectures by commercial vendors, systems integrators and industry standards organizations, the need for management standardization is great.

2.2.1 Consumer Electronics Manufacturer

For one example, take a typical problem faced daily by a large manufacturer of projection televisions-technical support. It is often easy to diagnose problems over the phone between a technician and the consumer. However, in those times when catastrophic failure of the television has created a condition that makes the problem indistinguishable for a consumer or technician, WSDM can alleviate some of the problems that this kind of situation creates.

Projection televisions are large and cumbersome and it may not be profitable or even realistic for a consumer to ship it back to the manufacturer. The current solution for most manufacturers is to have on-site repair by factory-trained and certified technicians. The cost to maintain and train these technicians is high and presents a huge overhead cost that not all manufacturers are willing to pay. So how can WSDM help in this situation?

WSDM does not create a web services client, server or service oriented architecture. WSDM simply provides the management framework upon which to build management applications. In the case of the projection television manufacturer, the manufacturer could service-enable and *"WSDM-enable"* its projection televisions before they are shipped from the manufacturing plant. The television manufacturer could then create management based applications that proactively diagnose problems instead of doing diagnosis in a reactionary mode.

For example, most projection televisions use lamps that eventually fail and have to be replaced by the consumer. It would be extremely cost beneficial as well as reputation-building to both the consumer and manufacturer to have the television "WSDM-enabled" to proactively diagnose the potential of failure of the consumer's television lamp and automatically ship the consumer a new lamp before it fails. In this respect, "WSDM-enabling" a device makes perfect sense.

2.2.2 Managing the Network

In another example of the potential application of WSDM, the arena of network management is high on the agenda of most organizations that manage IT infrastructures. Not all devices in the network management sphere have compatible protocols or even run the same application to manage their devices on the network. These include routers, load balancers, security gateways

as well as managed switches. What WSDM provides is a standard by which any of these devices can access management operations. It provides also a discovery as well as notification mechanism for these devices to automatically discover each other to share management related operations and data.

2.2.3 Managing Printers by the Local Systems Administrator

Most Systems Administrators are taxed with the burden of knowing multiple management applications, hardware and network applications; generally at the expense of on-the-job training by the employer. This does nothing but to cause frustration and management headaches to the Systems Admistrator. In an ideal world, all devices would have the same protocol, same network architecture, and produce the same results. However, all devices in the real world sometimes become single points of consternation because they only solve one issue needed by the consumer.

Printers are one example of this. Printers cannot manage themselves but they can be managed. They talk multiple languages to their printers such as PCL, PostScript, HPGL, and PCL-V. They all have management applications that enable either simple or complex management tasks such as alignment and print quality. However, every manufacturer's management application framework is different.

However, WSDM can give printers a framework to be single application management-enabled; either remotely or within a local network. This can enable the Systems Administrator to deploy profiles for printers and simply plug the printer into a network, instantly have it available to all users in the network and have it instantly discoverable to the management application of their choice. Manufacturers can benefit from WSDM-enabling devices by providing metrics on data sent to the printer. This could allow manufacturers to design printers for the way and manner they are used; based upon usage patterns, real-time and historical configurations and types of data being sent to printers.

2.2.4 Managing Web Services

Using WSDM, management applications can actually use web services to manage the very service oriented architectures upon which they're built. This is an actual standards extension of WSDM as a management framework also known as Management of Web Services (MOWS). This is discussed in detail in section 5.2, "Organization of the WSDM Standard".

Imagine a management framework that enables other web services to proactively diagnose and fix problems before they occur. This kind of scenario is especially important in service oriented architectures. Because there is not a single point of failure for this type of architecture, web services could fail without notifying other web services or devices on the network. So for a web service to proactively choose a web service based upon quality of service, latency, or actual jitter of the network transport makes a strong case for proactive management. WSDM enables web services to proactively manage each other; thereby decreasing the potential of failures in service oriented architectures and increasing the confidence of the actual consumers of web services.

3 The Objectives of WSDM

3.1 Architectural Origins of WSDM

WSDM was developed with a set of architectural foundations as its base; namely Web Services **[WS-Arch]** and Service Oriented (SOA) Architectures. WSDM itself is a specification and specification set, MUWS and MOWS, for managing devices as well as web services using web services; which are inherently dependent upon a Service Oriented Architecture foundation. The objectives of the specifications are many fold and these are described in the following sub-sections.

3.1.1 Resource Orientation

Historically, managers have accessed resources through management agents running on the resource. Some management agents support standard protocols for communication, like SNMP and WBEM, some are proprietary. In addition, the way that the agents communicate with the resource can be standard or proprietary. Managers accessing these resources would have to find the resource, find which agent was responsible for the resource, ask the agent for information about the resource or to make changes to the resource. This created a situation for managers where layers of discovery were being done as well as support for an ever growing number of protocols to agents. Besides resource access, many agents also provide many other useful services for managers.

By describing and offering resource access interfaces for RESOURCES directly rather than through intermediaries, WSDM makes resources Web services which can now participate directly in a service oriented architecture and business processes. It also allows the managers to concentrate on what resource they need to affect rather than having to keep track of which agents control, resources and protocol switching. The Web services infrastructure and bridges to agents take care of addressing and accessing the resource. This does not imply that agents are no longer part of the management infrastructure; agents are still providers of key management services and, as described in the next section, are excellent places to implement bridges to WSDM to provide integration for its resources into the WSDM SOA.

3.1.2 Implementation Isolation

Because WSDM is based on Web services, one is able to use the loose coupling features, platform agnostisism, and service orientation enabled by Web services to isolate manageable resources access from their manageable resource implementations. The clients' use of a manageable resource is consistent regardless of what implementation choices have been made. In fact, a resource's implementation can migrate over time from a bridge to provide immediate support to direct support by the resource without impacting management application and others who may be using the manageable resource. It can even migrate from Java to C implementation during this transition with no effect on the clients. Some common implementation topologies will be:

- Bridges to Agents to access all resources as WSDM manageable resources

- Proxies or adapters for the manageable resource which communicates in another protocol or native API to the resource. Proxies may be collocated with the resource or not. This allows existing resources to participate in WSDM without affecting their instrumentation.
- Direct support by a resource can be provided when the resource is capable of running a Web services stack or receiving, parsing, and responding to SOAP messages **[SOAP]**. Web services stacks can be surprisingly small; some are less than 40k bytes.

3.1.3 Composability of Services

WSDM needs to scale in several directions: small, constrained devices to large, sophisticated systems, as well as a few resources to hundreds of thousands of resources. In order to scale, the specification takes advantage of the composability of services afforded by Web services architectures. WSDM requires very few properties or operations; this means that the overhead of supporting WSDM is very low for small systems. However, WSDM provides a rich set of capabilities which can be used to provide descriptions of very sophisticated systems. This same composability allows implementers and deployers to compose in support for appropriate levels of qualities of service, like security, reliable messaging, brokered notifications, XPath **[XPath]** support, etc. This is a key feature of Web services and WSDM.

3.1.4 Model Agnostic

WSDM describes HOW to access management data pertaining to managed resources by means of a Web service protocol. This protocol makes use of management related concepts. These concepts serve to structure the management data into categories that would are useful to a management system. These categories, called manageability capabilities in WSDM, help to define a protocol that is uniquely useful to management applications.

On the other hand, WSDM itself does not define a data or information model. That is, it does not define the properties, operations, relationships, and events of managed resources. WSDM can be used to provide Web services interfaces for resources described by any resource generic model, such as CIM, SID, SNMP, or proprietary models. This feature is what we call model agnosticism. A goal of WSDM was to provide protocol specifically designed for management and imbedding management concepts while preserving independence of any particular model, including CIM, SID, or SNMP, of managed resources themselves. Thus, legacy application can be easily wrapped with a WSDM interface to provide Web services access to the model already used by the application.

3.1.5 Enabling Inspection

WSDM enables inspection (or discovery) of resource interfaces (properties, operations and events) at design time where there is no resource to ask directly, as well as at run time where finding and processing XML documents **[XML]** may be expensive. It is important that the managers are able to determine and integrate resource types and expected resources to manage without imposing that the resource already be installed and running. This is especially important where those resources are in the process of being installed, provisioned, and activated. Therefore, it's also necessary for managers to 'discover' new instances of resources in their environment and to attempt to understand and accommodate them as much as possible. For this reason, it's important that manageable resources not only support the Web services messages,

297 it's also important that there are WSDL **[WSDL]** and XML schemas **[XML Schema]** to describe
298 them that are accessible at design and run time.

4 What is WSDM?

The WSDM standard specifies how the manageability of a resource is made available to manageability consumers via Web services. This section describes the principal components of the WSDM architecture, its modes of interaction used and the facilities it provides in support of this architecture and interaction.

The focus of the WSDM architecture is the ***manageable resource***. The manageable resource must be represented as a Web service. In other words, management information regarding the resource must be accessible through a ***Web service endpoint.*** To provide access to a resource, this endpoint must be able to be referenced by an endpoint-reference, or ***EPR***, as defined in the WS-Addressing **[WS-Addressing]** standard. Endpoints that support access to manageable resources are called manageability endpoints. The implementation behind manageability endpoints must be capable of retrieving and manipulating the information related to a manageable resource.

The EPR provides the target location to which a ***manageability consumer*** directs messages. The manageable resource may also direct notifications of significant events to a manageability consumer, provided the consumer has subscribed to receive notifications. Thus, WSDM covers three modes of interaction between a manageable resource and a manageability consumer. These modes of interaction are as follows:

- A manageability consumer can retrieve management information about the manageable resource. For example, the consumer can retrieve the current operating status of the manageable resource or the current state of the process running on the manageable resource.
- A manageability consumer may affect the state of some manageable resource by changing its management information.
- A manageable resource may inform, or notify, a manageability consumer of a significant event. This mode of interaction requires the manageability consumer to subscribe to receive events on a desired topic.

WSDM is about defining a common manageability structure and message exchange format through which a manageable resource and a manageability consumer can *talk* to each other regardless of implementation or platform. Compliance with the WSDM standard requires that both manageable resource and manageability consumer are able to generate messages of the specified format and are able to accomplish certain resource management goals. Thus, WSDM defines a Web service message protocol for management information that may be shared by manageable resource and manageability consumer in a vendor-neutral, platform-independent manner.

A WSDM service is a management interface for a manageable resource present on the Web. However, WSDM, with some minor exceptions such as WSDM metrics, does not itself specify the content of any accessible management information. Only the format for retrieving and for manipulating management information is specified by the WSDM standard.

In summary, this section highlights the three components of the WSDM architecture, the manageable resource, the Endpoint Reference (EPR), and the manageability consumer and the three modes of interaction between these components.

The following sections describe the facilities used by WSDM to support this interaction among components. These facilities are the building blocks of the WSDM architecture.

4.1 The Resource Property Document

The XML representation of a manageable resource is described by an XML document called the ***resource properties document***. This document is referenced from a WSDL port type describing the operations of the Web service representing a manageable resource. For further information regarding the resource property document, see section 5.1, “The WSDM Technology Stack”, in this introduction.

A resource property exposes some information that is part of the state model for a manageable resource. The resource property document represents a logical composition of resource property elements presenting a view of the state of a manageable resource. The resource property document plays the central role in exposing the state of a manageable resource.. Construction of XML Schemas that describe the structure of resource property documents is covered extensively in the WSDM Primer.

4.2 Manageability Capabilities

A manageability capability is a set of properties, operations, and events, enabling a resource to be managed in a particular way. A manageable resource must be capable of supporting one or more capabilities that expose the manageable features of the resource. The manageability capabilities of a resource are part of the metadata describing the resource. Manageability capabilities are described in a resource property document associated with a manageable resource. Each capability is associated with a (possibly empty) set of properties, operations and events supported by a manageable resource and exposed via a Web service interface.

The WSDM-defined manageability capabilities and their supported facilities are as follow:

The **Identity** capability exposes the ResourceId of a manageable resource. A ResourceId is a globally unique identifier that is neither mutable nor modifiable. The ResourceId is correlatable: if two reported ResourceIds are identical, then they refer to the same manageable resource.

The **ManageabilityCharacteristics** capability exposes a list of ManageabilityCapability elements. Each element in the list denotes a capability supported by the manageable resource. Each list element is either a WSDM specified manageability capability or a capability that is specific to the type of manageable resource.

The **CorrelatableProperties** capability exposes a list of properties whose values are useful when determining whether two different ResourceIds from two different manageability providers actually refer to the same manageable resource.

The **Description** capability exposes the Caption, Description, and Version properties of the manageable resource.

The **State** capability exposes the current state and the last state transition of a manageable resource. The WSDM specification allows a resource to define its own state model. Support for this capability indicates that information about the state model of a manageable resource can be retrieved by a manageability consumer. For example, a resource may expose information about its current state and last state transition.

The **OperationalStatus** capability exposes the high-level health of a manageable resource from a simple operational perspective. The OperationalStatus property of a resource may have one of the following values: Available, PartiallyAvailable, Unavailable or Unknown.

The **Metrics** capability exposes metric information relevant to the performance and operation of a manageable resource. WSDM defines some metrics for a Web service resource and allows all resources to define any suitable and relevant metrics.

The **Configuration** capability exposes properties of the manageable resource that can be modified by the manageability consumer and which change the operation behavior of the resource.

The following list of capabilities is related to the management of a manageable resource. However, these capabilities may also be offered by endpoints other than manageability endpoints. For example, a registry endpoint may offer one or more of the following capabilities:

The **Relationships** capability exposes the relationships in which a resource participates. Facilities exposed by this capability include retrieving relationships, querying a resource for its participation in a specific type of relationship, and notifying on the creation or the deletion of a relationship in which the resource participates

The **RelationshipResource** capability exposes the properties of a manageable resource representing a relationship. These properties may include the name, type and role in the relationship

The **Advertisement** capability exposes a mechanism that emits a notification upon the creation or destruction of a manageable resource. See section 4.5, "Advertisement", for further discussion of this capability

A URI is used to represent each manageability capability. These URIs are specified in the WSDM specification and are considered opaque identifiers to the manageability consumer. A WSDM service must implement a minimum set of these capabilities. For example, the Identity capability must be implemented on every WSDM manageable resource. In addition to the WSDM defined capabilities, oother resource-specific capabilities may also be implemented.

4.3 Management Events

Management events are asynchronous notifications denoting a significant change of state in a manageable resource. In some cases the principal message exchange pattern between a manageable resource and a manageability consumer will take place via notifications rather than via the consumer requesting management information from the resource. Notifications and management events allow a manageability consumer to efficiently monitor the state of a manageable resource.

Each management capability (see section 4.2, "Manageability Capabilities") is associated with a topic on which notifications can be generated. For example, the topic associated with the State capability is StateCapability. WSDM defines additional topics for specific situations. Each topic in WSDM is associated with a message type of ManagementEvent. In other words, a message issued on a topic associated with a manageability capability has a precise format specified by the WSDM Event Format (WEF). The WEF contains an eventId, information regarding the source of the event, the reporter of the event, a message about what happened to the resource, and the date and time of the event. Notification messages contain additional information unique to the

specific event; for example, a state transition event includes Time, TransitionIdentifier, EnteredState, and PreviousState.

A manageability consumer subscribes to management events on the topics supported by the resource. In a subscription, the manageability consumer specifies the target resource, the topic of the desired event messages, any filtering that the service should provide when formatting the message for the subscription, and the desired form of delivery. A manageability consumer can later unsubscribe from the event topic.

4.4 Message Exchange Patterns

The WSDM specification defines a set of Message Exchange Patterns (MEPs) that support how a manageable resource and a manageability consumer may interact. MEPs may involve requests and responses exchanged between a manageability consumer and a manageable resource, for example within SOAP messages. MEPs are based upon the operations supported by a manageability interface. These supported operations are classified into three categories.

- Manageability consumer requests to the manageable resource for property information
- Manageability consumer commands to the manageable resource
- Manageability consumer subscriptions and manageable resource generated notifications.

4.4.1 Requests for Property Information

A request for property information relies upon a message exchange pattern (MEP) that draws from a certain set of operations on a manageability endpoint. This set of operations is as follows:

- The **GetResourceProperty** operation retrieves the value of a specified resource property for a manageable resource.
- The **GetMultipleResourceProperties** operation retrieves values for a specified set of resource properties for a manageable resource
- The **QueryResourceProperties** operation retrieves a portion of the resource properties document from a manageable resource using a query language such as XPath **[XPath]**
- The **QueryRelationshipsByType** operation retrieves information on a particular type of relationship in which the resource participates. This operation is defined in the WSDM specification

For examples of MEPs based upon these operations, see the appropriate WSDM specification or the WSDM Primer. Except as otherwise noted, all operations are defined in the WS-ResourceProperties specification upon which the WSDM specification is built. See section 5.1, "The WSDM Technology Stack" for additional information about the WS-ResourceProperties specification.

4.4.2 Commands to the Resource

A request to change the value of a resource property relies upon a message exchange pattern (MEP) that draws from a certain set of operations on a manageability endpoint. This set of operations is as follows:

- The **SetResourceProperties** operation takes resource properties as supplied by a manageability consumer and correspondingly modifies (inserts, updates, and/or deletes) the given properties in the resource property document for a manageable resource.

For examples of MEPs based upon these operations, see the appropriate WSDM specification or the WSDM Primer. Except as otherwise noted, all operations are defined in the WS-ResourceProperties specification upon which the WSDM specification is built. See section 5.1, "The WSDM Technology Stack" for additional information about the WS-ResourceProperties specification.

4.4.3 Subscriptions and Notifications

A request for a subscription relies upon a message exchange pattern (MEP) that draws from a certain set of operations on a manageability endpoint. This set of operations is as follows:

- The **Subscribe** operation requests that notifications be sent to a manageability consumer
- The **GetCurrentMessage** operation requests that a notification producer for a manageable resource return the last notification on a given topic
- The **PauseSubscription** operation requests a temporary hold on an existing subscription for a manageability consumer. A paused subscription continues to exist but does not propagate notifications during the period of time it is paused.
- The **ResumeSubscription** operation requests the re-activation of paused subscription for a manageability consumer.

In addition, the manageability consumer of Notifications may support the following operation:

- The **Notify** operation receives notifications on behalf of a manageability consumer

The WS-Notification family of standards also supports the distribution of notifications through intermediaries, such as messaging software. A NotificationBroker requires many of these operations as part of its interface. For example, a NotificationBroker is required to support the Subscribe and Notify operations. The WS-BrokeredNotification specification specifies MEPs that rely upon a certain set of operations that must be supported by the NotificationBroker. This set of operations is as follows:

- The **RegisterPublisher** operation creates the registration of a manageable resource as a notification publisher at a NotificationBroker
- The **Destroy** operation destroys the registration of a manageable resource at a NotificationBroker.

For examples of MEPs based upon these operations, see the appropriate WSDM specification or the WSDM Primer. Except as otherwise noted, all operations are defined in the WS-BaseNotification specification upon which the WSDM specification is built. See section 5.1, "The WSDM Technology Stack" for additional information about the WS-ResourceProperties specification.

4.5 Advertisement

The Advertisement capability constitutes a special type of notification. The Advertisement capability is exposed by a Web service to provide a notification on the creation or the destruction

of a manageable resource. Note that a Web service may offer this capability even though the Web service itself is not a manageable resource.

For example, a system might include a registry of resources. A registry entry is added as a resource is created. The corresponding registry entry is removed as a resource is destroyed. This registry of resources could then expose the creation and destruction information via the Advertisement capability. Through the Advertisement capability our example system would expose resource creation and destruction events with manageability consumers. A typical means of collecting diverse manageable resources into a registry is to use a service group as defined in the WS-ServiceGroup specification.

Notifications generated under the Advertisement capability are produced on the following topics:

- The creation of a manageability endpoint
- The creation of a manageable resource
- The destruction of a manageability endpoint
- The destruction of a manageable resource.

5 Structure of the WSDM Standard

The WSDM standard is built upon W3C and other OASIS standards. This section briefly summarizes the WSDM technology stack. It also describes the organization of the WSDM specification itself.

5.1 The WSDM Technology Stack

The eXtensible Markup Language (XML) is the fundamental technology underpinning the WSDM technology stack. All WSDM messages are serialized and transported as XML documents. The format of such a document is rigorously defined via XML Schema and respective standards. WSDM also uses XML Schema to define critical portions of the message exchange between a manageable resource and a manageability consumer.

The WSDM specification relies upon two fundamental Web services technologies, SOAP and WSDL. Since a key component of the WSDM architecture is the Endpoint Reference (EPR), the WSDM specification also relies upon the W3C WS-Addressing standard. SOAP messages directed to a manageable resource must conform to the EPR-to-SOAP bindings as described in the WS-Addressing standard.

WSDM uses WSDL to describe the interface provided by a manageability endpoint. The WSDM Primer provides examples defining the construction of WSDL documents describing how a manageability consumer can access the manageability capabilities of a manageable resource.

The next level of the WSDM technology stack consists of a set of OASIS standards specifying how to represent and access an XML representation of a resource. This set of standards is generally referred to as the WS-ResourceFramework. A WS-Resource is a Web service that exposes an XML representation of a resource. An external-facing Web service providing the manageability for a resource must be capable of providing information about the state of the resource.

Note that a manageable resource is also a WS-Resource. However, a WS-Resource is not necessarily a manageable resource. In other words, the WSDM standard extends the definition of WS-Resource by composing manageability capabilities with a WS-Resource.

The WS-ResourceFramework set of standards consists of the following standards:

- The WS-Resource **[WS-Resource]** specification specifies the basic notion of a WS-Resource. This specification should be read before beginning work with the WSDM standard
- The WS-ResourceProperties **[WS-ResourceSpecification]** specification defines the resource properties document. This specification should be read before beginning work with the WSDM standard.
- The WS-ResourceLifetime **[WS-ResourceLifetime]** specification specifies the interface for destroying a WS-Resource
- The WS-ServiceGroup **[WS-ServiceGroups]** specification defines how WS-Resources may be grouped in a set

- The WS-BaseFaults **[WS-BaseFaults]** specification defines a standard format for fault messages

The WS-ResourceProperties specification requires that every WS-Resource must be exposed through a resource property document. This is the reason why the resource property document is considered to be one of the fundamental building blocks of the WSDM standard. See section 4.1 "The Resource Property Document" for additional information about the resource property document. The WS-ResourceProperties specification also defines the relationship between a Web service endpoint and the resource property document for an underlying stateful resource. A resource property document must be described in the WSDL using XML Schema (type definition). The type of a resource property document is established by the root element of the XML Schema and is associated with a WSDL portType exposing the operations of a WS-Resource. This association is achieved via a WS-ResourceProperties defined attribute @ResourceProperties.

In the following example, *wsrf-rp* is the namespace prefix referring to the WS-ResourceProperties specification, *MyPdaDeviceProperties* is the root element of the XML Schema and *pda-prop* refers to the namespace defined by the XML schema. The following example describes a resource property document for a device called *MyPdaDevice:*

```
<portType name="MyPdaDevicePortType"
   wsrf-rp:ResourceProperties="pda-prop:MyPdaDeviceProperties">
```

A WS-Resource must implement a logical resource property document of the type declared with its WSDL portType.

A final set of OASIS specifications define the Message Exchange Patterns (MEPs) involved in subscriptions and notifications. These standards provide the following:

The WS-Topics **[WS-Topics]** specification defines how topics, on which subscriptions and notifications are based, are represented in XML. This specification should be read before beginning work with the WSDM standard

The WS-BaseNotification **[WS-BaseNotification]** specification defines the basic structure of subscription and notification messages. This standard should be read before beginning work with the WSDM standard

The WS-BrokeredNotification **[WS-BrokeredNotification]** specification defines an advanced architecture for subscriptions and notifications over Web services involving a third-party message recipient/distributor

In particular, the WSDM standard specifies additional topics dealing with management events to which manageability consumers may subscribe for a manageable resource. In addition, the WSDM standard augments the structure of an event contained in a notification with special features pertaining to the management of the event and of likely interest to a manageability consumer.

5.2 Organization of the WSDM Standard

The WSDM standard consists of two standards known as Management Using Web Services (MUWS) and Management of Web Services (MOWS).

The MUWS standard deals with the basic mechanisms and Message Exchange Patterns (MEPs) for managing any manageable resource ***using*** Web services as the platform for exchanging

messages. MUWS defines the manageability capabilities described in section 4.2, "Manageability Capabilities". MUWS itself is specified in two parts:

- MUWS part 1 **[MUWS Part 1]** addresses the fundamental capabilities of a manageable resource. These fundamental capabilities include resource identity, manageability characteristics, and correlatable properties. For more information on these MUWS capabilities, see section 5, “Capabilities” of the WSDM Primer. MUWS part 1 also provides a discussion of the WSDM Event Format (WEF). The interested reader is directed to MUWs part 1 for a detailed explanation of the MUWS architecture and principal concepts comprising the WSDM standard.
- MUWS part 2 **[MUWS Part 2]** addresses the set of WSDM manageability capabilities described in section 4.2, "Manageability Capabilities", other than those covered in MUWS part 1.

MUWS part 1 and MUWS part 2 each have an XML Schema and a WSDL definition. These documents should be incorporated into the definition of a manageable resource in order to make use of MUWS facilities.

The MOWS **[MOWS]** standard addresses the management of a Web service itself. The MOWS standard may be viewed both as an application of the WSDM MUWS standard and as an extension of the WSDM MUWS standard

In MOWS, a Web service is the manageable resource. Keep in mind that a Web service itself, as well as each process comprising the service, has state expressed by a state model as defined by Web Service Management: Service Lifecycle **[WSLC]**. Thus, a Web service can be a target of manageability by Web services. In addition to using the manageability capabilities and features of the MUWS standard, the MOWS standard introduces its own manageability capabilities and extends several MUWS capabilities to accommodate the special requirements of managing a Web service.

6 Summary

To summarize, WSDM is not a panacea for all things management-related; however it can provide a solid, standards-based framework for managing computing resources across the IT environment or consumer devices halfway across the world. WSDM builds upon standards rather than redefining or re-inventing technologies that already have strong industry footings. While it is ideal to show real world examples of how this technology can be implemented, over time the industry will shake out incompatibilities and inconsistencies within the WSDM standard through actual successive and quality-controlled implementations. For further information on WSDM, please reference the WSDM Primer on the OASIS home page as well as the base reference specifications upon which WSDM is built in section 7,"References". For further information, please contact the WSDM Technical Committee members as described in the status section from the cover page.

7 References

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

[MUWS Part 1] William Vambenepe, *Web Services Distributed Management:Management using Web Services (MUWS 1.0) Part 1*, OASIS Committee Draft, December 2004, http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf

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

[SOAP] Don Box, et al., *Simple Object Access Protocol (SOAP) 1.1*, W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[WS-Addressing] Don Box, et al., *Web services Addressing (WS-Addressing)*, W3CMember Submission, August 2004, http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/

[WS-Arch] David Booth, et al. *Web Services Architecture*, W3C Working Group Note, February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[WS-BaseFaults] Steve Tuecke, et. al., *Web Services Base Faults 1.2 (WS-BaseFaults)*, Oasis Working Draft, November, 2004, http://docs.oasis-open.org/wsrf/2004/11/wsrf-WS-BaseFaults-1.2-draft-03.pdf

[WS-BaseNotification] Steve Graham, et al., *Web Services Base Notification 1.2 (WS-BaseNotification)*, OASIS Working Draft, June 2004, http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf

[WS-BrokeredNotification] Dave Chappell, et al., *Web Services Brokered Notification 1.2 (WS-BrokeredNotification)*. OASIS Working Draft, July 2004l http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BrokeredNotification-1.2-draft-01.pdf

[WS-Resource] Steve Graham, et al. *Web Service Resource 1.2 (WS-Resource)*, OASIS Working Draft, October 2004, http://www.oasis-

open.org/apps/org/workgroup/wsrf/download.php/9547/wsrf-WS-Resource-1.2-draft-01.doc

[WS-ResourceLifetime] Latha Srinivasan, et al., Web Services Resource Lifetime 1.2 (WS-ResourceLifetime), OASIS Working Draft, June 2004, http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-03.pdf

[WS-ResourceProperties] Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-ResourceProperties)*, OASIS Working Draft, June 2004, http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-04.pdf

[WS-ServiceGroup] Tom Maguire, et al., *Web Services Service Group 1.2 (WS-ServiceGroup)*, OASIS Working Draft, June 2004, http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-02.pdf

[WS-Topics] William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, OASIS Working Draft, Jully 2004, http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf

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

[WSLC] Hao He, et al., *Web Service Management: Service Lifecycle*, W3C Note, February 2004, http://www.w3.org/TR/2004/NOTE-wslc-20040211/

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

[XML Schema] Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C Recommendation, May 2001, http://www.w3.org/TR/xmlschema-1/
Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C Recommendation, May 2001, http://www.w3.org/TR/xmlschema-2/

[XPath] James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C Recommendation, November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116

Appendix A. Acknowledgments

The following people made contributions to the WSDM MUWS Primer Version 1.0: Vaughn Bullard, Fred Carter, Mark Ellison, Heather Kreger, Richard Landau, Bryan Murray, Mitsunori Satomi, Thomas Studwell, Kirk Wilson, Zhili Zhang with special thanks to Vaughn Bullard, Bryan Murray, Kirk Wilson, and Mark Ellison as editors.

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

Appendix B. Revision History

Rev	Date	By Whom	What
CD	2006-02-24	Kirk Wilson	Title changes
Final wd	2006-01-27	Kirk Wilson	Prepare final wd for TC vote
wd v5	2006-01-27	Kirk Wilson	Add references and acknowledgements
Introduction v1	2005-09-06	Vaughn Bullard, Kirk Wilson, Bryan Murray	Extracted Chapter 1 from Primer; made stand alone as WSDM Primer: An introduction to WSDM. Added references to base specifications in Section 5 References. Added Real World Examples, new Overview and Summary.
wd-12	2005-08-30	Bryan Murray	Incorporate review of section 3, formatting changes throughout
wd-11	2005-08-26	Bryan Murray, Mark Ellison, Kirk Wilson, Heather Kreger	Modified sections as homework to people at the f2f, critical review of section 1, updates to Relationships
wd-10	2005-08-22	Bryan Murray	Changes from f2f
wd-09	2005-08-19	Bryan Murray, Kirk Wilson, Heather Kreger	Added content to many sections
wd-08	2005-07-20	Kirk Wilson	Sections 1, 3, 5
wd-07	2005-07-26	Bryan Murray	Updates from June 2005 f2f. Add Advertisement section.
wd-06	2005-06-29	Bryan Murray, Winston Bumpus, Kirk Wilson, Richard Landau	Formatting, FCAPS, section 4, introduction
wd-05	2005-05-04	Zhili Zhang	Add section 2.4 “Adding Property Access Operations”. Modify WSDL document in Appendix A.

Rev	Date	By Whom	What
wd-04	2005-04-27	Bryan Murray	Changes discussed during April f2f. Added text to Description section.
wd-03	2005-02-28	Bryan Murray	Outline revised based on Jan 2005 face-to-face, some section 2 content added
wd-02	2005-01-17	Bryan Murray	Revised outline to incrementally add capabilities
wd-01	2004-09-30	Bryan Murray	Initial version

724

Appendix C. Notices

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

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