

**Management Protocol Technical Committee**  
**Requirements List as of 08/21/2002**  
**Version 0.1**

*“Copyright (C) OASIS Open (2002). All Rights Reserved”*

**Management Web Service Requirements**

1. Focus on using web services to interface to CIMOM (as opposed to managing web services themselves which we could do in addition).
2. Use WSDL to describe the interface to the web services, as opposed to just supporting a SOAP protocol. This means the elements of the CIM meta model (classes, properties, etc) need to be mapped to WSDL elements.
3. Make sure that no capabilities of the CIM meta-model and CIM operations get lost on the way (particularly inheritance, associations, query language and indications).
4. Make sure that what we do will scale up to large number of CIM instances (e.g. >10000 per class).
5. The specification needs to be completely agnostic to any specific CIM schema / model. That means any future schema / model would need to be supported automatically.
6. We should do the specification in such a way that any code that implements the specification can be as schema/model agnostic as possible.
7. We will need to support future versions of the CIM operations. Example (not entirely out of the air because I have a CR pending on that): The Associators operation gets an additional argument specifying a query string to be applied to the target side objects. I suspect there are changes to CIM operations that require new versions of our specification. This again brings up the question how versions of our spec are synchronized with versions of CIM operations specs. Probably some procedural mechanism needs to be established.
8. The need to model all management protocol interactions over SOAP & WSDL
9. Use of XML for the data models
10. Transport protocol should be https for b2b, but otherwise should be transport agnostic
11. No forklift:
12. Co-exist with existing management infrastructure
13. Interoperate bi-directionally with existing management solutions in enterprise

14. Operational monitoring is only a small piece of management
15. Requires visibility into dependent components (such as services, process, servers, java component, .NET application, etc.)
16. Monitoring is not just operational data; need models of businesses
17. Performance monitoring
18. Business activity monitoring
- 19 . Articulate how web-service management relates to all the other web-service standards
  - WS-routing and WS-referral
  - WS-security and SAML
  - UDDI
20. Management messages can define changes to management (configuration of management)
21. Configuration messages should be able to specify their targets in a manner that is natural (web services might be specified by URL or WSDL file)
22. Configuration messages should include message content handling (the ability to, say, add instrumentation based on message content)
23. Management may be active, and should include capabilities for defining "ping" capabilities, etc.
24. Management infrastructure must not assume participation by or knowledge of the target services.
25. Management must be amenable to any services described by WSDL (This means that SOAP is not required.)
26. Configuration should be performed by messages, and not require restart or other human intervention.
27. Change Management: If different versions of the Web Services protocols are available over a period of time (SOAP 1.2, 2.0, etc.. and similarly for WSDL, and other protocols) we need a method/methodology to track version changes in a central piece of system.
28. The management protocol needs to be lightweight, extensible, simple to use, and have high performance in order to facilitate quicker adoption.

29. This TC is to strive to define a protocol for management of networks, systems, and web-services for interoperability and consolidation of protocol specific aspects of management.

30. Should support diverse management object models to accommodate various context specific management needs.

31. Should be model neutral, thus enabling a plug and play behavior and focusing on accommodation and consolidation of standards.

32. The overall focus should be towards the usage of industry standards like XML, SOAP, WSDL, UDDI, SSL encryption, HTTP, etc.

33. We also agree with concerns over security and we feel that it should be addressed from the onset.

34. We would also like to stress that there is a lot of work remaining to be done in the webservices, EAI and business process management space that could define the management protocol and the abstract interfaces/management operations thereof. Hence we feel that a sub-group, or a focus group within this TC would be an excellent idea.

### **Web Services Management Requirements**

1. Provide a way for a management application to identify the transactions provided by a web service (this is already done with WSDL)

2. Provide a standard way to express, for each transaction provided by a web service, the identity of managed objects that are required to process the transaction. I'm primarily interested in the identity of component web services, but the identity of other managed objects (computers, subsystems, etc) would be useful as well.

3. Provide a standard way for expressing a transaction trace, i.e. a list of the subsystems invoked by a transaction either as it is being processed, or once the transaction has completed.

4. Provide a standard way for expressing the availability of test transactions, i.e. transactions that are used to test the components of a distributed system.