ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 17963— 2016

Спецификация веб-служб для управления (WS-management)

(ISO/IEC 17963:2013, IDT)

Издание официальное

Москва
Стандартинформ
2016

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр» (ООО «ИАВЦ») на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 18 мая 2016 г. № 335-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 17963:2013 «Спецификация веб-служб для управления (WS-management)» (ISO/IEC 17963:2013 «Web Services for Management (WS-Management) Specification», IDT).

ИСО/МЭК 17963:2013 разработан подкомитетом ПК 38 «Платформы и сервисы для распределенных приложений» совместного технического комитета по стандартизации СТК 1 «Информационные технологии» Международной организации по стандартизации (ИСО) и Международной электротехнической комиссии (МЭК).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячном информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, 2016

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 2
3 Термины и определения 3
4 Сокращения и обозначения 4
5 Адресация 5
5.1 Адресация для управления 5
5.2 Версии адресации 13
5.3 Требования к совместимости 13
5.4 Использование адресации в WS-Management 15
6 Управляющие заголовки WS-Management 30
6.1 wsman:OperationTimeout 30
6.2 wsman:MaxEnvelopeSize 31
6.3 wsman:Locale 32
6.4 wsman:OptionSet 33
6.5 wsman:RequestEPR 36
7 Доступ к ресурсу 36
7.1 Общие положения 36
7.2 Унификация адресации 39
7.3 Операция Веб-службы Get 39
7.4 Операция Веб-сервиса Put 41
7.5 Delete 44
7.6 Создание ресурса (Create) 46
7.7 Доступ на уровне фрагмента 49
7.8 Get на уровне фрагмента 51
7.9 Put на уровне фрагмента 52
7.10 Delete на уровне фрагмента 55
7.11 Create на уровне фрагмента 55
8 Перечисление наборов данных 58
8.1 Общие положения 58
8.2 Перечисление (Enumerate) 60
8.3 Интерпретация фильтра 66
8.4 Pull 67
8.5 Release 72
8.6 Специальные запросы и перечисления на уровне фрагмента 73
8.7 Перечисление EPR 74
8.8 Renew 76
8.9 GetStatus 77
8.10 EnumerationEnd 78
9 Специализированные действия (методы) 79
10 Уведомления (обработка событий) 80
10.1 Общие положения 80
10.2 Подписка (Subscribe) 80
10.3 Получение статуса подписки (GetStatus) 99
10.4 Отказ от подписки (Unsibscribe) 100

10.5 Возобновление подписки (Renew) . . . 101
10.6 Окончание подписки SubscriptionEnd . . . 102
10.7 Подтверждение доставки . . . 103
10.8 Отказ от доставки . . . 105
10.9 Пропущенные события . . . 105
10.10 Управление доступом . . . 106
10.11 Рекомендации по реализации . . . 107
10.12 Анонсирование уведомлений . . . 107
11 Метаданные и обнаружение . . . 107
12 Безопасность . . . 110
12.1 Общие положения . . . 110
12.2 Профили безопасности . . . 110
12.3 Соображения безопасности для подписок событий . . . 111
12.4 Использование идентификационных данных при подписке . . . 112
12.5 Соотнесение событий с подпиской . . . 112
12.6 Ошибка аутентификации транспортного уровня . . . 113
12.7 Соображения о защищенности сторонних подписок . . . 113
13 Транспортные средства и кодирование сообщения . . . 113
13.1 SOAP . . . 113
13.2 Отсутствие ответа . . . 114
13.3 Повторное воспроизведение сообщений . . . 114
13.4 Пределы кодирования . . . 114
13.5 Двоичные вложения . . . 115
13.6 Чувствительность к регистру . . . 115
14 Отказы . . . 115
14.1 Введение . . . 115
14.2 Кодирование отказов . . . 116
14.3 Отказы NotUnderstood . . . 117
14.4 Вырожденные отказы . . . 118
14.5 Расширяемость отказов . . . 118
14.6 Основные отказы . . . 118
Приложение А (справочное) Условные обозначения . . . 137
Приложение В (обязательное) Соответствие . . . 138
Приложение С (обязательное) HTTP (S) транспорт и профиль безопасности . . . 139
Приложение D (справочное) Поддержка XPath . . . 145
Приложение Е (обязательное) Диалект фильтра селектора . . . 150
Приложение F (справочное) XML-схема для идентификации . . . 152
Приложение G (справочное) XML-схема и WSDL для операций доступа к ресурсу . . . 155
Приложение Н (справочное) XML-схема и WSDL операций перечисления . . . 161
Приложение I (справочное) XML-схема и WSDL операций уведомления . . . 172
Приложение J (справочное) XML-схема адресации . . . 182
Приложение К (справочное) XML-схема WS-Management . . . 186
Приложение L (справочное) Список изменений . . . 197
Приложение ДА (справочное) Сведение о соответствии ссылочного международного стандарта национальному стандарту . . . 198

Введение

Настоящий стандарт использует функциональность, подобную следующим Рекомендациям W3C:

- обработка событий веб-служб (WS-Eventing);
- передача данных веб-служб (WS-Transfer);
- перечисление веб-служб (WS-Enumeration).

Когда был определен WS-Management, данные Рекомендации W3C еще не существовали, и подобная функциональность была включена непосредственно в условия спецификации WS-Management. При последующих пересмотрах WS-Management эти функции могут быть включены посредством внешних ссылок на указанные Рекомендации W3C.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Спецификация веб-служб для управления (WS-management)

Web services for management (WS-Management) specification

Дата введения — 2017—06—01

1 Область применения

Спецификация веб-служб для управления (WS-management) описывает протокол Веб-служб, основанный на SOAP, для использования в предметных областях, связанных с управлением. Данные предметные области включают управление объектами такими, как персональные компьютеры, серверы, устройства, Веб-службы и другие приложения, а также другие объекты управления. Службы могут предоставлять отдельный интерфейс WS-Management или составной интерфейс, в котором интерфейс службы WS-Management объединен с некоторыми из многих других спецификаций Веб-служб.

Главное приложение таких служб находится в области управления системами. В целях продвижения функциональной совместимости между приложениями управления и управляемыми ресурсами, настоящий стандарт определяет базовый набор спецификаций Веб-служб и требований к использованию, которые представляют единый набор операций, составляющих ядро управления системами. Она включает возможность выполнить следующие действия:

- получить, обновить, создать и удалить отдельные экземпляры ресурсов, такие как параметры настройки и динамические значения;
- перечислить содержимое контейнеров и коллекций, таких как большие таблицы и журналы событий;
- подписаться на события, порождаемые управляемыми ресурсами;
- выполнять определенные методы управления со строго типизированными параметрами ввода и вывода.

В каждой из этих сфер области применения настоящий стандарт определяет минимальные требования к реализации совместимой Веб-службы. Реализация может не ограничиваться поддержкой только указанного набора операций, а также может принять решение не поддерживать одну или больше из указанных выше областей функциональности, если эта функциональность не соответствует целевому устройству или системе.

Настоящий стандарт предназначен для того, чтобы отвечать следующим требованиям:

- ограничить протоколы Веб-служб и форматы таким образом, чтобы было возможно создавать минимальные по размеру реализации Веб-служб как в аппаратных, так и в программных службах управления;
- определить минимальные требования к соответствию, не ограничивая реализации с более широким набором функций;
- гарантировать обратную совместимость и функциональную совместимость (интероперабельность) с версией 1.0 WS-Management;
- гарантировать компонуемость с другими спецификациями Веб-служб.

Издание официальное

2 Нормативные ссылки

Нормативные документы, полностью или частично упомянутые в настоящем стандарте, обязательны для их применения. Для датированных документов используют только указанное издание. Для недатированных документов используют самое последнее издание ссылочного документа (с учетом всех его изменений):

IETF RFC 2616, Р. Филдинг и др., Протокол передачи гипертекста (HTTP 1.1) (R. Fielding et al, Hypertext Transfer Protocol (HTTP 1.1)), Июнь 1999, gost.e-ecolog.ru

IETF RFC 2818, Э. Рескорла, HTTP поверх TLS (HTTPS) (E. Rescorla, HTTP over TLS (HTTPS)), Май 2000, gost.e-ecolog.ru

IETF RFC 3986, Т. Бернерс-Ли и др., Унифицированные идентификаторы ресурсов (URI): Общий синтаксис (T. Berners-Lee et al, Uniform Resource Identifiers (URI): Generic Syntax), Август 1998, http://gost.e-ecolog.ru .txt

IETF RFC 4122, П. Лич и др., Пространство имен URN для универсальных уникальных идентификаторов (UUID) (P. Leach et al, A Universally Unique Identifier (UUID) URN Namespace), Июль 2005, http://gost.e-ecolog.ru

IETF RFC 4178, Л. Жу и др., Простой и защищенный механизм согласования программного интерфейса общего сервиса безопасности (L. Zhu et al, The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism), Октябрь 2005, gost.e-ecolog.ru rfc4178.txt

IETF RFC 4559, К. Яганатан и др., Применение механизмов аутентификации Kerberos и NTLM, основанных на SPNEGO, для HTTP в Microsoft Windows (K. Jaganathan et al, SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows), Июнь 2006, gost.e-ecolog.ru

IETF RFC 5646, А. Филипс и др., Метки для идентификации языков (A. Phillips et al, Tags for Identifying Languages), Сентябрь 2009, gost.e-ecolog.ru

Директивы ISO/IEC, Часть 2. Правила, относящиеся к структуре и проектированию Международных стандартов (ISO/IEC Directives, Part 2: Rules for the structure and drafting of International Standards)

OASIS, А. Надалин и др., Профиль безопасности «Usertname Token» для Веб-служб 1.0 (A. Nadalin et al, Web Services Security Username Token Profile 1.0), Март 2004, gost.e-ecolog.ru oasis-200401-wss-username-token-profile-1.0.pdf

OASIS, А. Надалин и др., Безопасность Веб-служб: безопасность сообщений SOAP 1.0 (A. Nadalin et al, Web Services Security: SOAP Message Security 1.0) (WS-Security 2004), Март 2004, gost.e-ecolog.ru gost.e-ecolog.ru

OASIS, С. Андресон и др., Нотация для доверия в Веб-службах (WS-Trust) (S. Anderson et al, Web Services Trust Language (WS-Trust)), Декабрь 2005, gost.e-ecolog.ru

Стандарт Unicode версии 3.0 (The Unicode Consortium, The Unicode Standard Version 3.0), Январь 2000, gost.e-ecolog.ru

Консорциум «Юникод» Часто задаваемые вопросы по отметке порядка байтов (BOM) (The Unicode Consortium, Byte Order Mark (BOM) FAQ), gost.e-ecolog.ru

W3C, М. Гаджин и др., SOAP — Версия 1.2. Часть 1 — Основы обмена сообщениями (M. Gudgin, et al, SOAP Version 1.2 Part 1: Messaging Framework), Июнь 2003, gost.e-ecolog.ru

W3C, М. Гаджин и др., SOAP — Версия 1.2. Часть 2 — Дополнения (M. Gudgin, et al, SOAP Version 1.2 Part 2: Adjuncts), Июнь 2003, gost.e-ecolog.ru

W3C, М. Гаджин и др., Механизм оптимизации передачи сообщений SOAP (MTOM) (M. Gudgin, et al, SOAP Message Transmission Optimization Mechanism (MTOM)), Ноябрь 2004, gost.e-ecolog.ru TR/2004/PR-soap12-mtom-20041116/

W3C, Дж. Кларк и др., Язык путей в XML — Версия 1.0 (XPath 1.0) (J. Clark et al, XML Path Language Version 1.0 (XPath 1.0)), Ноябрь 1999, gost.e-ecolog.ru

W3C, Дж. Кован и др., Информационный набор XML — Вторая редакция (Инфонабор XML) (J. Cowan et al, XML Information Set Second Edition (XML Infoset)), Февраль 2004, gost.e-ecolog.ru TR/2004/REC-xml-infoset-20040204/

W3C, Г. Томпсон и др., XMLСхема — Часть 1. Структуры (XMLСхема 1) (H. Thompson et al, XML Schema Part 1: Structures (XML Schema 1)), Май 2001, gost.e-ecolog.ru

W3C, П. Бирон и др., XML Схема — Часть 2. Типы данных (XML Схема 2) (P. Biron et al, XML Schema Part 2: Data types (XML Schema 2)), Май 2001, gost.e-ecolog.ru

W3C, Адресация Веб-служб 1.0 — Базовая спецификация, Рекомендация W3C (Web Services Addressing 1.0 — Core, W3C Recommendation), Май 2006, gost.e-ecolog.ru

gost.e-ecolog.ru

W3C, Адресация Веб-служб 1.0 — Привязка SOAP, Рекомендация W3C (Web Services Addressing 1.0 — SOAP Binding, W3C Recommendation), Май 2006, gost.e-ecolog.ru

gost.e-ecolog.ru

W3C, Адресация Веб-служб 1.0 — Метаданные, Рекомендация W3C (Web Services Addressing 1.0 — Metadata, W3C Recommendation), Сентябрь 2007, gost.e-ecolog.ru

gost.e-ecolog.ru

W3C, Расширяемый язык разметки (XML) 1.0, Рекомендация W3C (Extensible Markup Language (XML) 1.0, W3C Recommendation), Октябрь 2000, gost.e-ecolog.ru

W3C, Пространства имен в XML, Рекомендация W3C (Namespaces in XML, W3C Recommendation), Январь 1999, gost.e-ecolog.ru

W3C, Э. Кристинсен и др., Язык описания Веб-служб, версия 1.1 (WSDL/1.1) (E. Christensen et al, Web Services Description Language Version 1.1 (WSDL/1.1)), Март 2001, gost.e-ecolog.ru

W3C, С. Боаг и др., XQuery 1.0: язык запросов XML (XQuery 1.0) (S. Boag et al, XQuery 1.0: An XML Query Language (XQuery 1.0)), Январь 2007, gost.e-ecolog.ru

3 Термины и определения

В настоящем стандарте применены следующие термины и определения. Наличие в тексте, не имеющего номер правила, таких нормативных терминов, как «должен быть», «не должен быть», «должен», «не должен», «может», или «не нужно», не означает, что текст не нормативен.

3.1 **может** (can): Используется для утверждений о возможности и возможности, будь то материальная, физическая или причинная.

3.2 **не может** (cannot): Используется для утверждений о возможности и возможности, будь то материальная, физическая или причинная.

3.3 **обусловлен** (conditional): Отмечает требования, которые следует строго соблюдать, чтобы соответствовать документу, когда указанные условия выполнены.

3.4 **обязательный** (mandatory): Отмечает требования, которые необходимо строго выполнять, чтобы соответствовать документу, и от которых не допускаются никакие отклонения.

3.5 **может** (may): Указывает на образ действия, допустимый в рамках конкретного документа.

3.6 **не требуется** (need not): Указывает на образ действия, допустимый в рамках конкретного документа.

3.7 **необязательный** (optional): Указывает на образ действия, допустимый в рамках конкретного документа.

3.8 **должен** (shall): Указывает на требования, которые необходимо строго соблюдать, чтобы соответствовать документу, и от которых не допускаются никакие отклонения.

3.9 **не должен** (shall not): Указывает на требования, которые необходимо строго соблюдать, чтобы соответствовать документу, и от которых не допускаются никакие отклонения.

3.10 **следует** (should): Указывает, что среди нескольких возможностей одна рекомендуется как особенно подходящая, не упоминая или исключая другие, либо подразумевая, что предпочитается, но не обязательно требуется определенный образ действия.

3.11 **не следует** (should not): Указывает, что определенная возможность или образ действия не поощряются, но не запрещаются.

3.12 **клиент** (client): Приложение, которое использует Веб-службы, определенные в данном документе, чтобы получить доступ к службе управления.

3.13 **потребитель** (customer): Веб-служба, которая запрашивает перечисление данных у источника данных.

3.14 **источник данных** (data sourse): Веб-служба, которая поддерживает перебор с использованием контекстов перечисления посредством операции Enumerate, определенной в настоящем стандарте.

3.15 **режим доставки** (delivery mode): Механизм, с помощью которого сообщения уведомления передаются от источника до приемника.

3.16 **контекст перечисления** (enumeration context): Контекст сессии, который представляет собой определенный перебор логической последовательности информационных элементов XML с использованием операции Pull, определенной в настоящем стандарте.

3.17 **приемник событий** (event sink): Веб-служба, которая получает уведомления.

3.18 **источник событий** (event source): Веб-служба, которая отправляет уведомления и принимает запросы о создании подписки.

3.19 **управляемый ресурс** (managed resource): Сущность, которая может представлять интерес для администратора. Это может быть физический объект, такой как портативный компьютер или принтер, либо абстрактный объект, такой как служба.

3.20 **уведомление** (notification): Сообщение, отправленное для указания того, что событие имело место.

3.21 **режим без запроса** (push mode):Механизм доставки, в котором источник отправляет приемнику сообщения о событиях как отдельные незапрашиваемые сообщения SOAP.

3.22 **ресурс** (resource): Веб-служба, адресуемая ссылкой оконечной точки, доступ к которой осуществляется посредством операций, определенных в настоящем стандарте. Такой ресурс может быть представлен XML-документом. XML-документ может являться представлением управляемого ресурса.

3.23 **класс ресурса** (resourse class): Абстрактное представление (тип) управляемого ресурса. Класс ресурса определяет представление связанных с управлением операций и свойств. Пример класса ресурса — описание операций и свойств для некоторого множества портативных компьютеров.

3.24 **фабрика ресурса** (resourse factory): Веб-служба, способная создавать новые ресурсы, используя операцию Create, определенную в настоящем стандарте.

3.25 **экземпляр ресурса** (resourse instance): Экземпляр класса ресурса.

Пример:

Набор связанных с управлением операций и значений свойств для определенного портативного компьютера.

3.26 **селектор** (selector): Связанная с ресурсом пара имя-значение, которая действует как дискриминант экземпляров, когда используется с моделью адресации WS-Management по умолчанию.

Селектор является, по существу, фильтром или «ключом», определяющим желаемый экземпляр ресурса. Селектор может не присутствовать, когда используются модели адресации, специфичные для служб.

Отношения служб к классам и экземплярам ресурсов:

- служба состоит из одного или более классов ресурса;
- класс ресурса может содержать ноль или более экземпляров.

Если существует более одного экземпляра для класса ресурса, они изолируются или идентифицируются посредством частей адреса SOAP для ресурса, таких как поля ResourceURI и SelectorSet в модели адресации по умолчанию.

3.27 **служба** (service): Приложение, которое предоставляет клиентам услуги управления, предоставляя доступ к Веб-службам, определенным в данном документе.

Как правило, служба эквивалентна сетевому «слушателю», связана с физическим транспортным адресом и является по существу типом точки доступа управляемости (manageability access point).

3.28 **подписчик** (subscriber): Веб-служба, которая отправляет запросы, чтобы создавать, возобновлять и/или удалять подписки.

3.29 **менеджер подписки** (subscribtion manager): Веб-служба, которая принимает запросы на управление, получение статуса, возобновление и/или удаление подписки от имени источника события.

4 Сокращения и обозначения

В настоящем стандарте применены следующие обозначения и сокращения.

4.1 BNF — форма Бэкуса-Наура.

4.2 BOM — отметка порядка байтов.

4.3 CQL — язык запросов CIM.

4.4 EPR — ссылка на оконечную точку.

4.5 GSSAPI — универсальный прикладной программный интерфейс служб безопасности.

4.6 SOAP — простой протокол доступа к объектам.

4.7 SPNEGO — простой и защищенный механизм переговоров GSSAPI.

4.8 SQL — язык структурированных запросов.

4.9 URI — универсальный идентификатор ресурса.

4.10 URL — унифицированный указатель ресурсов.

4.11 UTF — формат преобразования UCS.
4.12 UUID — универсально уникальный идентификатор.
4.13 WSDL — язык описания веб-служб.
4.14 WS-Man — веб-служба для управления.

5 Адресация

WS-Management опирается на механизм адресации, основанный на SOAP (например, определенный в 5.1), для определения ссылок на оконечные точки на других Веб-служб и определения некоторых заголовков, используемых в сообщениях SOAP. Данный механизм адресации семантически эквивалентен и полностью бинарно совместим с версией WS-Addressing, на которую ссылаются в WS-Management 1.0, поэтому данное изменение для WS-Management полностью обратно совместимо с существующими реализациями WS-Management.

5.2 определяет, как можно использовать более одной версии адресации с WS-Management, таких как версия, определенная в 5.1, или версия адресации из Рекомендации W3C. В настоящем стандарте, если конкретная версия не указана явно, термин «Адресация» относится в целом к любой версии адресации, как определено в 5.2.

Различные модели адресации могут использоваться с любой из версий адресации, описанных в 5.2. Реализации могут реализовывать любую из следующих моделей адресации:

- базовая адресация, как определено в 5.1;
- модель адресации по умолчанию, как определено в 5.4.2;
- новые модели адресации, которые не определены в настоящем стандарте. Эти модели адресации могут ввести дополнительные ограничения или требования для адресации.

5.1 Адресация для управления

Характеристики, определенные в данном пункте, обеспечивают механизм для адресации Веб-служб и сообщений, не зависящий от транспортного механизма. А именно, данный пункт определяет элементы XML для определения оконечных точек Веб-службы и обеспечения сквозной идентификации оконечной точки в сообщениях. Это позволяет системам обмена сообщениями поддерживать передачу сообщений через сети, включающие узлы с обработкой сообщений, такие как системы управления оконечными узлами, брандмауэры и шлюзы, нейтральным по отношению к транспорту способом.

5.1.1 Введение

Данный пункт определяет две взаимодействующие конструкции, ссылки оконечной точки и информационные заголовки сообщения, передающие информацию, которая, как правило, предоставляется транспортными протоколами и системами обмена сообщениями. Данные конструкции преобразуют эту информацию от нижележащих уровней в единый формат, который может быть обработан независимо от транспортного протокола или приложения.

Оконечная точка Веб-службы представляет собой сущность, процессор или ресурс, на которые можно сослаться и которые могут служить адресатом для сообщений Веб-службы. Ссылки оконечной точки содержат информацию, необходимую для идентификации и ссылки на оконечную точку Веб-службы, они могут использоваться несколькими различными способами:

- ссылки оконечной точки подходят для передачи информации, необходимой для получения доступа к оконечной точке Веб-службы;
- ссылки оконечной точки также используются для предоставления адресов отдельным сообщениям, посланным в Веб-службы и полученным от них.

Для второго варианта использования данный пункт определяет группу информационных заголовков сообщений, которые позволяют проводить унифицированную адресацию сообщений, независимую от нижележащего транспортного протокола. Данные информационные заголовки передают характеристики сообщения, не меняющиеся в течение всего маршрута, включая адреса оконечных точек источника и назначения, а также идентификатор сообщения.

Пример:

Следующий пример иллюстрирует использование этих механизмов в сообщении SOAP 1.2, посланном от gost.e-ecolog.ru ***к*** gost.e-ecolog.ru.

Строки с (002) по (014) представляют заголовок сообщения SOAP, в котором используются механизмы, определенные в данном пункте. Тело представлено строками от (015) до (017).

Строки с (003) по (013) содержат блоки информационных заголовков сообщения. Именно, строки с (003) по (005) определяют идентификатор для этого сообщения, строки с (006) по (008) определяют оконечную точку, в которой сообщение порождено, и строки с (009) по (011) определяют оконечную точку, в которую следует отправить ответ на данное сообщение, как ссылку оконечной точки. Строка (012) задает URI адреса окончательного приемника данного сообщения. Строка (013) задает ActionURI, определяющий ожидаемую семантику.

```
(001) <S:Envelope xmlns:S="gost.e-ecolog.ru
        xmlns:wsa="gost.e-ecolog.ru                                    >
(002)  <S:Header>
(003)   <wsa:MessageID>
(004)    uuid:6B29FC40-CA47-1067-B31D-00DD010662DA
(005)   </wsa:MessageID>
(006)   <wsa:From>
(007)    <wsa:Address>gost.e-ecolog.ru                                 >
(008)   </wsa:From>
(009)   <wsa:ReplyTo>
(010)    <wsa:Address>gost.e-ecolog.ru                                 >
(011)   </wsa:ReplyTo>
(012)   <wsa:To>gost.e-ecolog.ru                                  >
(013)   <wsa:Action>gost.e-ecolog.ru                                  >
(014)  </S:Header>
(015)  <S:Body>
(016)   ...
(017)  </S:Body>
(018) </S:Envelope>
```

5.1.2 Ссылки оконечной точки

Данный пункт определяет синтаксис ссылки оконечной точки (EPR).

5.1.2.1 Формат ссылок оконечной точки

Данный пункт определяет представление XML для ссылки оконечной точки как тип XML (wsa:EndpointReferenceType) и как элемент XML (<wsa:EndpointReference>).

Тип wsa:EndpointReferenceType используется везде, где ссылаются на оконечную точку Веб-службы. Следующий пример описывает содержимое такого типа:

```
<wsa:EndpointReference>
<wsa:Address>xs:anyURI</wsa:Address>
<wsa:ReferenceProperties>... </wsa:ReferenceProperties>?
<wsa:ReferenceParameters>... </wsa:ReferenceParameters>?
<wsa:PortType>xs:QName</wsa:PortType> ?
<wsa:ServiceName PortName="xs:NCName"?>xs:QName</wsa:ServiceName>?
<wsp:Policy> ... </wsp:Policy>*
</wsa:EndpointReference>
```

Ниже описаны атрибуты и элементы, перечисленные в этой схеме сообщения:

wsa:EndpointReference

Представляет собой некоторый элемент типа wsa:EndpointReferenceType. Данный пример использует предварительно определенный элемент <wsa:EndpointReference>, однако может использоваться любой элемент типа wsa:EndpointReferenceType.

wsa:EndpointReference/wsa:Address

Данный обязательный элемент (типа xs:anyURI) определяет URI адреса, задающего оконечную точку. Данный адрес может быть логическим адресом или идентификатором оконечной точки службы.

wsa:EndpointReference/wsa:ReferenceProperties/

Данный необязательный элемент содержит любое число отдельных свойств ссылки, связанных с оконечной точкой, предназначенных облегчить конкретное взаимодействие. Свойства ссылки — это элементы XML, которые требуются для того, чтобы должным образом взаимодействовать с оконечной точкой. Свойства ссылки предоставляются стороной, создавшей ссылку оконечной точки, и считаются непрозрачными для всех остальных обрабатывающих приложений.

Примечание — Использование свойств ссылки устарело; вместо них следует использовать параметры ссылки.

wsa:EndpointReference/wsa:ReferenceProperties/ {любой}

Каждый дочерний элемент элемента ReferenceProperties представляет отдельное свойство ссылки.

wsa:EndpointReference/wsa:ReferenceParameters/

Данный необязательный элемент содержит любое число отдельных связанных с оконечной точкой параметров, предназначенных облегчать конкретное взаимодействие. Параметры ссылки — элементы XML, необходимые для того, чтобы взаимодействовать должным образом с оконечной точкой. Параметры ссылки также предоставляются стороной, создавшей ссылку оконечной точки, и считаются непрозрачными для всех остальных обрабатывающих приложений.

см. в 5.4 некоторые параметры ссылки, специфичные для WS-Management.

wsa:EndpointReference/wsa:ReferenceParameters/ {любой}

Каждый дочерний элемент элемента ReferenceParameters представляет отдельный параметр ссылки.

wsa:EndpointReference/wsa:PortType

Данный необязательный элемент (типа xs:QName) определяет значение основного portType передаваемой оконечной точки.

Примечание — Использование wsa:PortType устарело.

wsa:EndpointReference/wsa:ServiceName

Данный необязательный элемент (типа xs:QName) задает определение <wsdl:service>, содержащее описание WSDL оконечной точки, на которую указывает ссылка. Имя службы обеспечивает связь с полным описанием оконечной точки службы. Необязательное неквалифицированное имя определяет конкретный порт в службе, которая соответствует оконечной точке.

Примечание — Использование wsa:ServiceName устарело.

wsa:EndpointReference/wsa:ServiceName/@PortName

Данный необязательный атрибут (типа xs:NCName) задает название определения <wsdl:port>, соответствующее оконечной точке, на которую указывает ссылка.

wsa:EndpointReference/wsp:Policy

Данный необязательный элемент определяет политику, которая относится к взаимодействию с оконечной точкой.

Примечание — Использование wsp:Policy устарело.

wsa:EndpointReference/{любой}

Механизм расширения, позволяющий определять дополнительные элементы.

wsa:EndpointReference /@ {любой}

Механизм расширения, позволяющий определять дополнительные атрибуты.

Пример:

Следующий пример иллюстрирует ссылку оконечной точки. Данный элемент дает ссылку на URI “gost.e-ecolog.ru ***:***

<wsa:EndpointReference xmlns:wsa=“...” xmlns:fabrikam=“...”>

<wsa:Address>gost.e-ecolog.ru ***>***

</wsa:EndpointReference>

5.1.2.2 Привязка ссылок оконечной точки

Когда необходимо адресовать сообщение к оконечной точке, информация, содержащаяся в ссылке оконечной точки, отображается на сообщение согласно преобразованию, которое зависит от протокола и представления данных, используемых для отправки сообщения. Определенные для протокола отображения (или привязки) определяют, как информация из ссылки оконечной точки копируется в поля сообщения и протокола. Данный пункт определяет привязку SOAP для ссылок оконечной точки. Это отображение может быть явным образом заменено другими привязками (определенными как привязки WSDL или как политики); однако в отсутствие применимой политики, определяющей, что должно использоваться иное отображение, должна применяться привязка SOAP, определенная в данном пункте. Чтобы обеспечить функциональную совместимость для широкого диапазона устройств, все реализации, соответствующие настоящему стандарту, должны поддерживать привязку SOAP.

Привязка SOAP для ссылок оконечной точки определяется следующими двумя правилами:

R5.1.2.2-1 Элемент wsa:Address в ссылке оконечной точки должен быть скопирован в поле заголовка wsa:To сообщения SOAP.

R5.1.2.2-2 Каждый элемент ReferencePropertyи ReferenceParameter становится блоком заголовка в сообщении SOAP. Каждый элемент ReferenceProperty или ReferenceParameter (включая все его дочерние элементы, атрибуты и пространства имен в области видимости) должны быть добавлены как блок заголовка в новом сообщении.

Пример:

Следующий пример показывает, как используется привязка SOAP по умолчанию для ссылок оконечной точки для конструирования сообщения, адресованное к оконечной точке:

```
<wsa:EndpointReference xmlns:wsa="..." xmlns:fabrikam="...">
<wsa:Address>gost.e-ecolog.ru >
<wsa:ReferenceParameters>
<fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
<fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
</wsa:ReferenceParameters>
</wsa:EndpointReference>
```

Согласно правилам отображения, сформулированным выше, значение адреса скопировано в заголовок «To», и элемент «CustomerKey» должен быть скопирован посимвольно как заголовок в сообщение SOAP, адресованное этой оконечной точке. Сообщение SOAP будет выглядеть следующим образом:

```
<S:Envelopexmlns:S="gost.e-ecolog.ru
    xmlns:wsa="..." xmlns:fabrikam="... ">
<S:Header>
  ...
<wsa:To>gost.e-ecolog.ru >
<fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
<fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
   ...
</S:Header>
<S:Body>
   ...
</S:Body></S:Envelope>
```

5.1.3 Информационные заголовки сообщения

Данный пункт определяет синтаксис информационного заголовка сообщения.

Информационные заголовки сообщения совокупно изменяют сообщение заголовками, показанными на рисунке 1. Эти заголовки обеспечивают идентификацию и местоположение оконечных точек, вовлеченных во взаимодействие. Основной шаблон взаимодействия, из которого составлены все другие, — это «в одну сторону». В этом шаблоне источник отправляет сообщение в место назначения без дальнейшего определения взаимодействия.

«Запрос-ответ» является общим шаблоном взаимодействия, который состоит из первоначального сообщения, отправленного исходной оконечной точкой (запрос), и последующего сообщения, отправленного из места назначения запроса назад к источнику (ответ). Ответ может быть сообщением приложения, отказом или любым другим сообщением.

Блоки информационного заголовка сообщения обеспечивают характеристики сообщения, не изменяющиеся в ходе доставки сообщения и представимые как целостные единицы. Информация в этих заголовках неизменна и не предназначена для изменения по пути следования сообщения.

Рисунок 1 показывает содержимое блоков информационных заголовков сообщения:

<wsa:MessageID> xs:anyURI </wsa:MessageID>

<wsa:RelatesTo RelationshipType="..."?>xs:anyURI</wsa:RelatesTo>

<wsa:To>xs:anyURI</wsa:To>

<wsa:Action>xs:anyURI</wsa:Action>

<wsa:From>ссылка-на-оконечную-точку</wsa:From>

<wsa:ReplyTo>ссылка-на-оконечную-точку</wsa:ReplyTo>

<wsa:FaultTo>ссылка-на-оконечную-точку</wsa:FaultTo>

Рисунок 1 — блоки информационных заголовков сообщения

Далее описаны атрибуты и элементы, перечисленные на рисунке 1:

wsa:MessageID

Данный необязательный элемент (типа xs:anyURI) однозначно определяет сообщение во времени и пространстве. Данный элемент должен присутствовать, если присутствуют wsa:ReplyTo или wsa:FaultTo. Никакие два сообщения с определенным намерением приложения не могут иметь общее значение wsa:MessageID. Сообщение может быть передано далее для любой цели (включая отказ соединения) и может использовать тоже самое значение wsa:MessageID. Значение данного заголовка — непрозрачный URI; для значения в настоящем стандарте не определена интерпретация помимо эквивалентности. Если ожидается ответ, это свойство должно присутствовать.

wsa:RelatesTo

Данный необязательный (повторяющийся) элемент указывает, как сообщение относится к другому сообщению в форме пары URI-QName. Дочерний элемент данного элемента (который имеет типа xs:anyURI) содержит wsa:MessageID связанного с ним сообщения или следующий известный URI, который означает «неуказанное сообщение»:

gost.e-ecolog.ru

Ответное сообщение должно содержать заголовок wsa:RelatesTo, состоящий из wsa:Reply и значения wsa:MessageIDсообщения-запроса.

wsa:RelatesTo / @RelationshipType

Данный необязательный атрибут (типа xs:QName) передает тип связи как QName. В случае, если он отсутствует, подразумеваемое значение этого атрибута — wsa:Reply.

В настоящем стандарте определен один тип связи, как показано в таблице 1.

Таблица 1 — Тип связи

QName	Описание
wsa:Reply	Указывает, что это ответ на сообщение, идентифицируемое определенным URI

wsa:ReplyTo

Данный необязательный элемент (типа wsa:EndpointReferenceType) предоставляет ссылку оконечной точки, которая определяет приемник, предназначенный для ответов на сообщение. Данный элемент должен присутствовать, если ожидается ответ. Если присутствует данный элемент, то должен присутствовать wsa:MessageID. Если ответ ожидается, то сообщение должно содержать заголовок wsa:ReplyTo. Отправитель должен использовать содержимое wsa:ReplyTo, чтобы сформулировать от-

ветное сообщение, как определено в 5.1.3.1. Если заголовок wsa:ReplyTo отсутствует, то содержимое заголовка wsa:From может использоваться для того, чтобы сформулировать сообщение к источнику. Данный заголовок может отсутствовать, если у сообщения нет содержательного ответа.

wsa:From

Данный необязательный элемент (типа wsa:EndpointReferenceType) предоставляет ссылку на оконечную точку, где возникло сообщение.

wsa:FaultTo

Данный необязательный элемент (типа wsa:EndpointReferenceType) предоставляет ссылку оконечной точки, которая определяет приемник, предназначенный для отказов, связанных с сообщением. Если присутствует данный элемент, то должен присутствовать wsa:MessageID. При формулировании сообщения об отказе, как определено в 5.1.3.1, отправитель должен использовать содержимое данного заголовка, чтобы сформулировать сообщение отказа. Если данный заголовок отсутствует, отправителю следует использовать содержимое заголовка wsa:ReplyTo, чтобы сформулировать сообщение-отказ. Если отсутствуют оба заголовка wsa:FaultTo и wsa:ReplyTo, отправитель может использовать содержимое заголовка wsa:From для формулирования сообщения об отказе.

wsa:To

Данный обязательный элемент (типа xs:anyURI) обеспечивает адрес приемника, предназначенного для этого сообщения.

wsa:Action

Данный обязательный элемент (типа xs:anyURI) однозначно определяет семантику, подразумеваемую сообщением. Рекомендуется, чтобы значение этого заголовка было URI, определяющий входные данные, результаты, или сообщение-отказ в рамках типа порта WSDL. Действие может быть явно или неявно связано с соответствующим определением WSDL. Наконец, если в дополнение к заголовку wsa:Action в запросе будет закодирован URI действия SOAP, то URI действия SOAP должен либо совпадать с значением, определенным в заголовке wsa:Action, либо быть установленными.

Выбор обработчика входящих сообщений основан на двух свойствах сообщения. Обязательные заголовки wsa:To и wsa:Action определяют целевое местоположение обработки и глагол или намерение сообщения.

Поскольку в настоящее время широко используется ряд сетевых технологий (например, NAT, DHCP и брандмауэры), многие реализации не могут назначать значимый глобальный URI для заданной оконечной точки. Чтобы позволить этим «анонимным» конечным точкам начинать шаблоны обмена сообщениями и получать ответы, адресация определяет следующий известный URI для использования конечными точками, у которых не может быть стабильного, разрешимого URI:

gost.e-ecolog.ru

Запросы, у которых заголовки wsa:ReplyTo, wsa:From и/или wsa:FaultTo используют данный адрес, должны обеспечивать некоторый механизм, находящийся вне настоящего стандарта, для доставки ответов или отказов (например, путем возврата ответа через то же самое транспортное соединение). Таким механизмом может быть простой транспортный протокол запроса/ответа (например, HTTP GET или POST). Данный URI может использоваться в качестве заголовка wsa:To для ответных сообщений и не должен использоваться в качестве заголовка wsa:To при других обстоятельствах.

5.1.3.1 Формулировка ответного сообщения

Ответ на запрос, удовлетворяющий спецификации адресации, должен быть построен согласно правилам, определенным в данном пункте.

Примеры:

1 Следующий пример иллюстрирует сообщение-запрос, использующий блоки информационных заголовков сообщения в сообщении SOAP 1.2:

```
<S:Envelope xmlns:S="gost.e-ecolog.ru
 xmlns:wsa="gost.e-ecolog.ru
 xmlns:f123="gost.e-ecolog.ru                    >
<S:Header>
<wsa:MessageID>uuid:aaaabbbb-cccc-dddd-eeee-ffffffffffff
```

</wsa:MessageID>
<wsa:ReplyTo>
<wsa:Address>http://business456.example/client1</wsa:Address>
</wsa:ReplyTo>
*<wsa:To S:mustUnderstand="1">*gost.e-ecolog.ru *>*
*<wsa:Action>*gost.e-ecolog.ru *>*
</S:Header>
<S:Body>
<f123:Delete>
<maxCount>42</maxCount>
</f123:Delete>
</S:Body>
</S:Envelope>

2 Следующий пример иллюстрирует ответное сообщение, использующее блоки информационных заголовков сообщения в сообщении SOAP 1.2:

<S:Envelope
*xmlns:S="*gost.e-ecolog.ru
*xmlns:wsa="*gost.e-ecolog.ru
xmlns:f123="http://www.fabrikam123.example/svc53">
<S:Header>
<wsa:MessageID>
uuid:aaaabbbb-cccc-dddd-eeee-wwwwwwwwwww
</wsa:MessageID>
<wsa:RelatesTo>
uuid:aaaabbbb-cccc-dddd-eeee-fffffffffff
</wsa:RelatesTo>
<wsa:To>
http://business456.example/client1
</wsa:To>
*<wsa:Action>*gost.e-ecolog.ru *>*
</S:Header>
<S:Body>
<f123:DeleteAck/>
</S:Body>
</S:Envelope>

5.1.3.2.1 Связывание действий с операциями WSDL

Для связи действия с входными данными, результатами и отказами в пределах типа порта WSDL адресация определяет два механизма: явную ассоциацию и шаблон действия по умолчанию.

5.1.3.2.2 Явная ассоциация

Действие может быть ассоциировано явным образом посредством атрибут wsa:Action.

Пример:

Рассмотрим следующий фрагмент WSDL:

*<definitions targetNamespace="*gost.e-ecolog.ru *.....*
<portType name="StockQuotePortType">
<operation name="GetLastTradePrice">
<input message="tns:GetTradePricesInput"
*wsa:Action="*gost.e-ecolog.ru *>*
<output message="tns:GetTradePricesOutput"
*wsa:Action="*gost.e-ecolog.ru *>*

```
</operation>
</portType>
...
</definitions>
```

Действие над входными данными операции GetLastTradePrice в StockQuotePortType явно определено как gost.e-ecolog.ru . Действие для выходных данных этой той же самой операции — gost.e-ecolog.ru .

5.1.3.2.3 Шаблон действия по умолчанию

В отсутствие атрибута wsa:Action следующий шаблон используется для конструирования действия по умолчанию для входных и выходных данных. Общая форма URI следующая:

targetNamespace/portTypeName/(inputName|outputNname)

«/» литеральный символ, который будет включен в действие. Значения свойств следующие:

- *targetNamespace* целевое пространство имен (/definition/@targetNamespace). Если *target namespace* заканчивается на «/» , то дополнительный «/» не добавляется;
- *portTypeName* название типа порта (/definition/portType/@name);
- (*inputName*|*outputName*) название элемента, как определено в Разделе 2.4.5 спецификации WSDL 1.1.

Для сообщений отказа данный образец не применяется. Вместо этого следующий URI является URI действия по умолчанию для сообщения об отказе:

gost.e-ecolog.ru

Пример:

Рассмотрим следующий фрагмент WSDL:

```
<definitions targetNamespace="gost.e-ecolog.ru ...>
...
<portType name="StockQuotePortType">
<operation name="GetLastTradePrice">
<input message="tns:GetTradePricesInput" name="GetQuote"/>
<output message="tns:GetTradePricesOutput" name="Quote"/>
</operation>
</portType>
...
</definitions>
```

targetNamespace = gost.e-ecolog.ru

portTypeName = StockQuotePortType

inputName = GetQuote

outputName = Quote

Применяя предыдущий шаблон с этими значениями, получаем следующее:

- действие для входных данных = gost.e-ecolog.ru ;
- действие для выходных данных = gost.e-ecolog.ru .

WSDL определяет правила для имени по умолчанию для входных или выходных данных, если атрибут имени не присутствует. Рассмотрим следующий пример:

Пример:

Ниже приведен фрагмент WSDL:

```
<definitions targetNamespace="gost.e-ecolog.ru ...>
...
<portType name="StockQuotePortType">
<operation name="GetLastTradePrice">
<input message="tns:GetTradePricesInput"/>
```

```
<output message="tns:GetTradePricesOutput"/>
  </operation>
  </portType>
  ...
  </definitions>
  targetNamespace = gost.e-ecolog.ru
  portTypeName = StockQuotePortType
```

Согласно правилам, определенным в подпункте 2.4.5 спецификации WSDL, если атрибут имени отсутствует для сообщения с входными данными операции вида запрос — ответ, значение по умолчанию — имя операции с добавленной к нему строкой «Request»

inputName = GetLastTradePriceRequest

Аналогично по умолчанию имя для выходных данных состоит из имени операции с добавленной строкой «Response».

outputName = GetLastTradePriceResponse

Применяя предыдущий пример с этими значениями, получаем следующее:

- действие для входных данных =

gost.e-ecolog.ru ;

- действие для выходных данных=

gost.e-ecolog.ru .

5.2 Версии адресации

Для поддержания совместимости с реализациями предыдущих версий WS-Management, данный протокол поддерживает обработку сообщений, сформированных в соответствии с предыдущими версиями. Однако WS-Management 1.1 также допускает опциональное использование рекомендации W3C WS-Addressing.

Для ясности и краткости используются следующие сокращения:

- «WSMA» обозначает версию Management Addressing, определеную в 5.1;
- «WSA-Rec» обозначает рекомендацию W3C WS-Addressing;
- «WS-Man 1.0» обозначает спецификацию WS-Management 1.0 и реализации, совместимые с указанной спецификацией;
- «WS-Man 1.1» обозначает данную спецификацию и реализации, совместимым с данной спецификацией;

«Анонимный URI адресации» относится к анонимному URI, который определен версией Адресации, использующейся в настоящее время. В WSA-Rec определен анонимный URI

gost.e-ecolog.ru .

В WSMA определен анонимный URI gost.e-ecolog.ru .

Примечание — Часть информации в данном разделе является рекомендацией клиентам относительно алгоритмов эффективного обмена информацией с неизвестными службами. Такую информативную рекомендацию не следует трактовать как нормативные требования к поведению клиентов или служб, соответствующих стандарту.

5.2.1 Технические различия

Спецификации WSMA и WSA-Rec ссылаются на различные пространства имен XML. Оконечная точка, посылающая сообщения Веб-службы, должна использовать для заголовков адресации SOAP одно пространство имен или другое; конечная точка получения может распознать одно пространство имен или оба пространства имен. Существующие реализации WS-Man 1.0 распознают только пространства имен WSMA. Взаимодействия между реализациями WS-Man 1.0 и WS-Man 1.1 должны будут допускать эти ограничения.

5.3 Требования к совместимости

Чтобы максимизировать функциональную совместимость реализаций WS-Management, клиенты и службы WS-Man 1.0 и WS-Man 1.1 должны быть в состоянии обмениваться сообщениями. Эти требования сведены в таблицу 2.

Таблица 2 — Требования функциональной совместимости (интероперабельность)

Требования функциональной совместимости между версиями WS-Management	Служба WS-Man 1.0	Служба WS-Man 1.1
Клиент WS-Man 1.0	Работает	Клиент WS-Man 1.0 должен быть в состоянии получить доступ к службе WS-Man 1.1, но могут потребоваться некоторые согласования
Клиент WS-Man 1.1	Клиент WSMan 1.1 должен быть способен получить доступ к службе 1.0	Работает, но могут потребоваться некоторые согласования

Однородные соединения клиентов и служб, соответствующих стандарту (то есть клиента версии 1.0 со службой версии 1.0 или клиента версии 1.1 со службой версии 1.1) могут обмениваться сообщениями в соответствии со своим версиями спецификации. Чтобы гарантировать надежную связь, разнородные соединения должны отвечать определенным требованиям и осуществлять определенные упорядочивающие стратегии.

В частности клиенты и службы, которые реализуют WS-Man 1.0, могут использовать только WSMA в любых обменах; поэтому все обмены с оконечными точками версии 1.0 используют только WSMA. Это заключение отражено в таблице 3.

Таблица 3 — Версии WSA в обменах

Совместимая версия адресации	Служба WS-Man 1.0	Служба WS-Man 1.1
Клиент WS-Man 1.0	WSMA	WSMA
Клиент WS-Man 1.1	WSMA	WSMA или WSA-Rec

5.3.1 Обнаружение или согласование

Если можно определить возможности обслуживания для клиента относительно WSA, такое обнаружение более эффективно, чем согласование версии WSA. Например, если служба поддерживает Indentify, то клиент может определить заранее протокол WS-Man, а также версию или версии адресации, поддерживаемые службой. Поэтому поддержка Identify является обязательной в настоящем стандарте в случае, когда используется WSA-Rec.

Identify может использоваться следующим образом:

- клиент отправляет службе сообщение Identify;
- если служба не поддерживает Identify, клиент может прийти к заключению, что служба является реализацией WS-Man 1.0 и поддерживает только WSMA;
- если служба успешно обрабатывает сообщение Identify, клиент определяет версии Адресации по элементу AddressingVersionURI (как определено в разделе 11) и может выбрать соответствующую версию;
- если ответное сообщение Identify не содержит версий Адресации, то для клиента нет никакого способа узнать, какую версию Адресации использовать, и ему придется использовать одну из стратегий, описанных в 5.3.2.

В любом случае, чтобы избежать ненужного повторного обнаружения или согласования, клиенту WS-Man 1.1 следует сохранять информацию о возможностях оконечных точек служб, где это применимо.

5.3.2 Стратегии согласования клиента

Клиент WS-Man 1.0 будет использовать в обменах сообщениями только WSMA. Клиент WS-Man 1.1, однако, может использовать в обменах сообщениями либо WSMA, либо WSA-Rec. Если клиент WS-Man 1.1 не знает версию WSA службы, он может использовать различные стратегии при первом установлении связи со службами. Клиент может начать обмен сообщениями с любой версии WSA, используя WSA-Rec или WSMA в сообщении запроса. Обмен сообщениями может продолжиться следующим образом:

- тип стратегии 1: клиент отправляет запрос, используя WSA-Rec. Заголовки SOAP WSA-Rec должны быть отмечены атрибутом mustUnderstand = “1”, чтобы гарантировать, что если приемник не

поддерживает версию адресации WSA-Rec, будет сгенерирован отказ. Тогда клиент может повторить операцию, используя WSMA;

- тип стратегии 2: клиент отправляет запрос, используя WSMA, и службы WS-Man 1.0 и службы WS-Man 1.1 отвечают на запрос, используя WSMA.

5.3.3 Инициирование обменов сообщениями

Исходящие сообщения, инициированные реализацией WS-Man, должны использовать ту же самую версию адресации, которая использовалась в ссылке оконечной точки, в которую отправляют эти сообщения, например, если сообщение-запрос Subscribe использует WSA-Rec в заголовках SOAP (например, для wsa:To и wsa:ReplyTo), но если использует WSMA для NotifyTo, то ответ на Subscribe отправят, используя WSA-Rec, но события отправят, используя WSMA.

5.3.4 Нормативные правила

R5.3.4-1 Если служба WS-Man 1.1 поддерживает WSA-Rec, то она должна также поддерживать операцию Identify.

R5.3.4-2 Служба WS-Man 1.1 должна поддерживать WSMA и должна поддерживать WSA-Rec.

R5.3.4-3 Реализация WS-Man 1.1 должна отправлять сообщения в оконечные точки, используя ту же самую версию Адресации, которая используется в ссылке оконечной точки (Endpoint Reference) оконечной точки назначения (см. 5.2).

R5.3.4-4 В рамках единственного сообщения SOAP реализация WS-Man 1.1 должна использовать одну и ту же самую версию адресации для всех заголовков адресации SOAP.

Поскольку WS-Man 1.1 допускает использование любой версии Адресации, R5.3.4-4 устраняет возможность смешивания этих двух версий для заголовков SOAP WSA, но не запрещает, чтобы ссылки оконечной точки, которые могут присутствовать в других местах сообщения, имели другую версию.

Для обеспечения пути миграции от WSMA до WSA-Rec схема некоторых сообщений допускает EndpointReferenceType любой используемой версии. В то время как сама схема написана очень общим способом (то есть, используя xs:any), что позволяет включать любой произвольный XML, реализации должны ограничивать содержимое этого элемента до одного из типов ссылки оконечной точки.

Примечание — Это позволяет существующим реализациям WS-Man 1.0 соответствовать стандарту, одновременно обеспечивая более новым реализациям путь миграции. В новейших реализациях настоятельно рекомендуется поддерживать обе версии адресации.

5.4 Использование адресации в WS-Management

Данный пункт описывает использование ссылок оконечной точки независимо от того, использует ли реализация адресацию WS-Management (см. 5.1) или версию рекомендации W3C WS-Addressing.

Адресация (при любом типе адресации) ссылки оконечной точки (EPR) используется для передачи информации, необходимой, чтобы адресовать оконечную точку Веб-службы. WS-Management определяет модель адресации по умолчанию, которая может использоваться в EPR.

5.4.1 Использование ссылок оконечной точки

WS-Management использует EPR в качестве механизма адресации экземпляров ресурса. WS-Management также определяет модель адресации по умолчанию для использования при адресации к ресурсам. В случаях, где эта модель адресации по умолчанию неприменима, как, например, в системах с определенными моделями адресации, или когда ссылка оконечной точки предоставлена службой обнаружения, службы могут использовать свои специфические модели адресации, если они основаны на любой версии адресации, поддерживаемой WS-Management.

R5.4.1-1 Все сообщения, адресованные к классу или экземпляру ресурса, на который ссылается EPR, должны выполнять правила адресации для предоставления содержания EPR (адрес и параметры ссылки) в сообщении SOAP. Это правило также относится к сообщениям, таким как Pull или Release, которые продолжают операцию, начатую в предыдущем сообщении. Несмотря на то, что такие сообщения содержат контекстную информацию, которая связывает их с предыдущей операцией, информация EPR должна присутствовать в сообщении для маршрутизации к правильному обработчику.

Правило R5.4.1-1 разъясняет, что для сообщений таких, как Pull или Renew, требуется полный EPR. Для Pull, например, данный EPR должен совпадать с ERP исходной операции Enumerate, несмотря на то, что EnumerateResponse возвращает объект контекста, который, казалось бы, устранил потребность в EPR. EPR все еще требуется, чтобы маршрутизировать сообщения должным образом. Точно также запрос Renew использует SubscriptionManager EPR, полученный в SubscribeResponse.

Когда служба включает EPR в ответном сообщении, она должна быть готова принять последующие сообщения-запроса, предназначенные этому EPR для того же самого отдельного управляемого ресурса. Клиенты не обязаны обрабатывать или модифицировать EPR, предоставленные им службой, перед тем, как использовать их для обращения к управляемому ресурсу.

R5.4.1-2 EPR, возвращенный службой, должен быть приемлемым для данной службы, чтобы обращаться к тому же самому управляемому ресурсу.

R5.4.1-3 Все EPR, возвращенные службой, выраженные с использованием модели адресации WS-Management по умолчанию (см. 5.4.2) или с помощью какой-либо другой модели адресации, должны быть действительными, пока существует управляемый ресурс.

5.4.2 Модель адресации WS-Management по умолчанию

WS-Management по умолчанию определяет модель адресации для ресурсов. Служба не обязана использовать эту модель адресации, но она подходит для многих новых реализаций и может увеличить вероятность успешного взаимодействия между клиентами и службами.

Настоящий стандарт использует примеры этой модели адресации, содержащие составные части — заголовки SOAP ResourceURI и SelectorSet. Данная спецификация не зависит от фактической модели данных и не определяет структуру ResourceURI или набор значений селекторов для заданного ресурса. Они могут быть определены поставщиком или другими спецификациями.

Описание и использование данной модели адресации в настоящем стандарте не означает, что поддержка этой модели адресации обязательна для совместимой службы.

Весь нормативный текст, примеры и правила соответствия в 5.4.2 и 5.4.2.2 предполагают, что служба основана на модели адресации по умолчанию. В случаях, когда эта модель адресации не используется, данные правила не применяются.

Модель адресации по умолчанию использует представление EPR, которое является последовательностью следующих заголовков SOAP:

- **wsa:To** (обязательный заголовок): транспортный адрес службы;
- **wsman:ResourceURI** (обязательный заголовок, если используется модель адресации по умолчанию): URI представления класса ресурса или представления экземпляра;
- **wsman:SelectorSet** (необязательный заголовок): заголовок, который определяет или «выбирает» экземпляр ресурса, к которому необходимо получить доступ, если существует более одного экземпляра класса ресурса.

Во всех сообщениях, использующих модель адресации по умолчанию, значение wsman:Resource URI должно быть отмечено атрибутом s:mustUnderstand, установленным в значение «true». В противном случае служба, которая не понимает данную модель адресации, может непреднамеренно возвратить ресурс, не запрашиваемый клиентом.

Модель адресации WS-Management по умолчанию определена в следующей схеме XML для EPR:

```
(1)  <wsa:EndpointReference>
(2)    <wsa:Address>
(3)      Сетевой адрес
(4)    </wsa:Address>
(5)    <wsa:ReferenceParameters>
(6)      <wsman:ResourceURI> URI ресурса</wsman:ResourceURI>
(7)      <wsman:SelectorSet>
(8)        <wsman:Selector Name="selector-name"> *
(9)          Значение селектора
(10)       </wsman:Selector>
(11)     </wsman:SelectorSet> ?
(12)   </wsa:ReferenceParameters>
(13) </wsa:EndpointReference>
```

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsa:Address

URI транспортного адреса

wsa:ReferenceParameters/wsman:ResourceURI

URI класса ресурса или экземпляра, к которому запрашивается доступ.

Как правило, данный URI представляет класс ресурса, но может представлять и экземпляр. Комбинация этого URI и wsa:ToURI составляет полный адрес класса ресурса или экземпляра.

wsa:ReferenceParameters/wsman:SelectorSet:

необязательное множество селекторов, как описано в 5.4.2.2.

Эти значения используются для выбора экземпляра, если ResourceURI определяет цель с несколькими экземплярами.

Когда в сообщении SOAP используется модель адресации по умолчанию, адресация определяет преобразование и разворачивание заголовков.

Пример:

Ниже приведен пример определения EPR:

```
(1)  <wsa:EndpointReference>
(2)    <wsa:Address>Адрес</wsa:Address>
(3)    <wsa:ReferenceParameters xmlns:wsman="...">
(4)      <wsman:ResourceURI>resURI</wsman:ResourceURI>
(5)      <wsman:SelectorSet>
(6)        <wsman:Selector Name="имя-селектора">
(7)          значение-селектора
(8)        </wsman:Selector>
(9)      </wsman:SelectorSet>
(10)   </wsa:ReferenceParameters>
(11)  </wsa:EndpointReference>
```

Это определение адреса при использовании в сообщении SOAP преобразуется следующим образом: wsa:Address становится wsa:To, и разворачиваются и сочетаются параметры ссылки. Следующий пример показывает типовое сообщение SOAP, использующее WSMA:

```
(1) <s:Envelopexmlns:wsa="gost.e-ecolog.ru                                    ">
(2)  <s:Header>
(3)  <wsa:To>Адрес</wsa:To>
(4)  <wsa:Action> URI действия</wsa:Action>
(5)  <wsman:ResourceURI s:mustUnderstand="true">resURI</wsman:ResourceURI>
(6)  <wsman:SelectorSet>
(7)  <wsman:Selector Name="имя-селектора">
(8)  значение-селектора
(9)  </wsman:Selector>
(10) </wsman:SelectorSet>
(11) ...
(12) </s:Header>
(13) <s:Body> ... </s:Body>
(14) </s:Envelope>
```

Следующее сообщение показывает типовое сообщение SOAP, использующее WS-Rec:

```
(1) <s:Envelope xmlns:wsa="gost.e-ecolog.ru                                    >
(2) <s:Header>
(3) <wsa:To s:mustUnderstand="true">Адрес</wsa:To>
(4) <wsa:Action s:mustUnderstand="true"> URI действия</wsa:Action>
(5) <wsman:ResourceURI s:mustUnderstand="true"
(6) wsa:isReferenceParameter="true">resURI</wsman:ResourceURI>
```

(7) <wsman:SelectorSet wsa:isReferenceParameter="true">

(8) <wsman:Selector Name="имя-селектора">

(9) значение-селектора

(10) </wsman:Selector>

(11) </wsman:SelectorSet>

(12) ...

(13) </s:Header>

(14) <s:Body> ... </s:Body>

(15) </s:Envelope>

В обоих случаях элементы wsa:To, wsman:ResourceURI и wsman:SelectorSet работают вместе для *ссылки* на экземпляр ресурса, которым будут управлять, но фактический *метод* или *операция*, которые должны быть выполнены сданным ресурсом, обозначены в заголовке wsa:Action.

Пример:

Ниже приведен пример заголовков адресации, основанных на модели адресации по умолчанию, в фактическом сообщении:

(1) <s:Envelope

*(2) xmlns:s="*gost.e-ecolog.ru *03/05/soap-envelope"*

*(3) xmlns:wsa="*gost.e-ecolog.ru

*(4) xmlns:wsman="*gost.e-ecolog.ru *>*

(5) <s:Header>

(6) ...

*(7) <wsa:To>*gost.e-ecolog.ru *>*

(8) <wsman:ResourceURI s:mustUnderstand="true">

(9) gost.e-ecolog.ru

(10) </wsman:ResourceURI>

(11) <wsman:SelectorSet>

(12) <wsman:Selector Name="LUN"> 2 </wsman:Selector>

(13) </wsman:SelectorSet>

(14) <wsa:Action> gost.e-ecolog.ru

</wsa:Action>

(15) <wsa:MessageID> urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a91

</wsa:MessageID>

(16) ...

(17) </s:Header>

(18) <s:Body> ... </s:Body>

(19) </s:Envelope>

В этом примере используются следующие определения:

wsa:To

адрес сети (или транспортного уровня) службы

wsman:ResourceURI

ResourceURI класса ресурса или экземпляра ресурса, к которому будет осуществлен доступ

wsman:SelectorSet

обертка для селекторов

wsman:SelectorSet/wsman:Selector определяет или выбирает экземпляр ресурса, к которому будет осуществляться доступ, если существует более одного экземпляра ресурса.

В этом случае селектором является «LUN» (Logical Unit Number, номер логической единицы) и выбранное устройство — блок (unit) номер «2».

wsa:Action определяет, какая операция должна быть выполнена с ресурсом (в этом случае, «Get»).

wsa:MessageID

определяет данное конкретное сообщение уникальным образом для прослеживания и соотнесения с ответом. В примерах в настоящем стандарте часто используется формат, определенный в RFC 4122, но он не является обязательным.

5.4.2.1 ResourceURI

ResourceURI используется для того, чтобы указывать на класс или экземпляр ресурса.

R5.4.2.1-1 На формат wsman:ResourceURI не накладываются ограничения, за исключением требований RFC 3986.

Формат и синтаксис ResourceURI могут быть любыми URI, корректными согласно RFC 3986. Хотя по умолчанию схема URI не задана, http: и urn: широко используются по умолчанию. Если используется http:, пользователи могут ожидать, что по этому адресу будет находиться Веб-ресурс документации. Элементы wsa:To и wsman:ResourceURI работают вместе для фактического определения целевого ресурса.

R5.4.2.1-2 URI, специфичный для поставщика или для организации, должен содержать доменное имя Интернета в первой последовательности символов после схемы, такое как «example.org» в ResourceURI в следующем примере.

Пример:

(20) <s:Header>

(21) <wsa:To> gost.e-ecolog.ru ***</wsa:To>***

(22) <wsman:ResourceURI>

(23) http//schemas.example.org/2005/02/hardware/physDisk

(24) </wsman:ResourceURI>

(25) ...

(26) </s:Header>

R5.4.2.1-3 При использовании модели адресации по умолчанию требуется параметр ссылки wsman:ResourceURI в сообщениях со следующими wsa:ActionURI:

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

В следующих сообщениях требуется, чтобы EPR был возвращен в элементе SubscriptionManager сообщения SubscribeResponse. Формат EPR определяется службой и может включать или не включать ResourceURI:

gost.e-ecolog.ru

gost.e-ecolog.ru

Несмотря на то, что модель адресации WS-Management по умолчанию требует наличие заголовка SOAP ResourceURI, он может быть короткой и очень простой формы, такой как gost.e-ecolog.ru или gost.e-ecolog.ru .

R5.4.2.1-4 Для сообщения — запроса специализированных действий (методов) в сообщении может присутствовать заголовок ResourceURI, чтобы помочь маршрутизировать сообщение к правильному обработчику.

R5.4.2.1-5 Элемент ResourceURI не следует включать в другие сообщения, такие как ответы или события, если связанный элемент EPR не включает его в свой элемент ReferenceParameters.

На практике элемент wsman:ResourceURI требуется только в тех запросах, в которых идет ссылка на целевой класс ресурса. Ответы не адресованы к ресурсу управления, таким образом, wsman:ResourceURI в этом контексте не имеет смысла.

R5.4.2.1-6 Когда используется модель адресации по умолчанию и элемент wsman:ResourceURI отсутствует или выражен в неправильной форме, служба должна вернуть сообщение-отказ wsa:DestinationUnreachable со следующим кодом детализации

gost.e-ecolog.ru

R5.4.2.1-7 Элемент wsman:ResourceURI должен использоваться только для того, чтобы указывать на идентификацию ресурса, он не может использоваться для указания на действие, применяемое к тому ресурсу, что выражается должным образом посредством wsa:Action URI.

В специализированных методах, основанных на WSDL, присутствует как идентичность ResourceURI для адресации, так и wsa:ActionURI для выполнения. Во многих случаях ResourceURI — просто псевдоним для идентичности WSDL и порта, и wsa:Action URI — конкретный метод этого порта (или интерфейса).

Теоретически для определения экземпляра ресурса в случае множественных экземплярах может быть достаточно единственного URI, тем не менее рекомендуется для определения местонахождения службы WS-Management использовать элемент wsa:To, для определения класса ресурса использовать элемент wsman:ResourceURI, а для ссылки на экземпляр ресурса использовать элемент wsman:SelectorSet. Если ресурс состоит из единственного экземпляра, то элемент wsman:ResourceURI указывает на единственный экземпляр.

Этот сценарий использования является не строгим требованием, а рекомендацией. Служба может использовать отдельные селекторы для любой заданной операции, даже применять к тому же самому классу ресурса, и может разрешить или потребовать селекторы для операции Enumerate.

См. рекомендации по унификации адресации в 7.2.

У специализированных действий есть две различные идентичности: ResourceURI, которое может идентифицировать WSDL и порт (или интерфейс), и wsa:Action URI, который идентифицирует конкретный метод. Если в интерфейсе существует только один метод, в некотором смысле ResourceURI и wsa:Action URI идентичны.

Не является ошибкой использовать wsa:Action URI для ResourceURI специализированного метода, однако оба они требуются в сообщении для унификации обработки клиентами и службами.

Пример:

Следующее действие для перезагрузки сетевой платы может использовать следующее EPR:

(1) <s:Header>

(2) <wsa:To>

(3) gost.e-ecolog.ru

(4) </wsa:To>

(5) <wsman:ResourceURI>gost.e-ecolog.ru

</wsman:ResourceURI>

(6) <wsa:Action>

(7) gost.e-ecolog.ru

(8) </wsa:Action>

(9) ...

(10)</s:Header>

Во многих случаях ResourceURI эквивалентен имени и порту WSDL, и wsa:Action URI содержит необязательный символ как суффикс, что показано в следующем примере.

Пример:

(1) <s:Header>

(2) <wsa:To>

(3) gost.e-ecolog.ru

(4) </wsa:To>

(5) <wsman:ResourceURI>gost.e-ecolog.ru

</wsman:ResourceURI>

(6) <wsa:Action>

(7) gost.e-ecolog.ru

(8) </wsa:Action>

(9) ...

(10) </s:Header>

Наконец, ResourceURI может быть абсолютно не связан с wsa:Action URI, как в следующем примере.

Пример:

(1) <s:Header>

*(2) <wsa:To>*gost.e-ecolog.ru >

(3) <wsman:ResourceURI>

(4) gost.e-ecolog.ru

(5) </wsman:ResourceURI>

(6) <wsa:Action>

(7) gost.e-ecolog.ru

(8) </wsa:Action>

(9) ...

(10)</s:Header>

Все эти варианты правомерны.

При использовании с подписками EPR, описанные элементами wsa:Address и wsman:ResourceURI (и дополнительно значениями wsman:SelectorSet), определяют источник события, к которому направлена подписка. Во многих случаях ResourceURI определяет реальный или виртуальный журнал событий, и подписка предназначена для того, чтобы предоставить уведомления в реальном времени о любых новых записях, добавленных в журнал. Во многих случаях элемент wsman:SelectorSet не может использоваться в качестве части EPR.

5.4.2.2 Селекторы

В модели адресации WS-Management по умолчанию селекторы — это необязательные элементы, используемые для определения экземпляров в пределах класса ресурса. Для таких операций, как Get или Put, селекторы используются для определения единственного экземпляра класса ресурса, на который ссылается ResourceURI.

На практике, поскольку ResourceURI часто выступает как таблица или «класс», элемент SelectorSet используется для различения, определения конкретного «ряда» или «экземпляра». Если ResourceURI подразумевает только один экземпляр класса ресурса, то SelectorSet может быть пропущен, так как ResourceURI выступает как полный идентификатор ресурса. Если требуется более одного значения селектора, вся группа селекторов интерпретируется службой для обращения к определенному экземпляру. Селекторы интерпретируются, как будто они объединены подразумеваемыми логическими операторами И.

В некоторых информационных доменах значения, на которые ссылаются селекторы, служат «ключами», являющимися частью содержимого самого ресурса, тогда как в других доменах селекторы — часть логической или физической системы каталогов или пространства поиска. В этих случаях селекторы используются для определения ресурса, но не являются частью представления.

R5.4.2.2-1 Если ресурс имеет более одного экземпляра, может использоваться элемент wsman:SelectorSet, чтобы различать, какой экземпляр является целевым при использовании модели адресации WS-Management по умолчанию. С элементом wsman:SelectorSet может появиться любое число значений wsman:Selector, необходимых для точного определения экземпляра класса ресурса. Служба может учитывать регистр имен селектора и значений (см. 13.6), как требуется в нижележащей среде выполнения.

Если клиенту необходимо обнаружить политику интерпретации регистра значений селектора, служба может предоставить документы метаданных, которые описывают эту политику. Формат таких метаданных выходит за рамки настоящего стандарта.

R5.4.2.2-2 Все содержимое в рамках элемента SelectorSet нужно рассматривать как единственный параметр ссылки с областью применения, относящейся к ResourceURI.

R5.4.2.2-3 Служба, использующая модель адресации WS-Management по умолчанию, должна исследовать все селекторы в сообщении и обработать их, как если бы они были логически связаны оператором И. Если множество селекторов оказывается некорректным для целевого экземпляра ресурса, следует вернуть клиенту отказ wsman:InvalidSelectors со следующими кодами детализации:

если селекторы отсутствуют:

- gost.e-ecolog.ru

если значения селектора некорректного типа:

- gost.e-ecolog.ru

если тип значения селектора правилен с точки зрения типов XML, но значение находится вне диапазона или иным образом неправомерно в определенном информационном домене:

- gost.e-ecolog.ru

если имя не является известным именем селектора:

- gost.e-ecolog.ru

R5.4.2.2-4 Атрибут SelectorNamene должен повторяться на том же самом уровне вложения. Если это происходит, службе следует вернуть отказ wsman:InvalidSelectors со следующим кодом детализации: gost.e-ecolog.ru

Данная спецификация не налагает требования на использование селекторов. Некоторые реализации могут решить использовать сложные схемы URI, в которых сам ResourceURI неявно определяет конкретный экземпляр.

Формат элемента SelectorSet следующий:

(1) <s:Envelope

(2) xmlns:s="gost.e-ecolog.ru

(3) xmlns:wsa="gost.e-ecolog.ru ***"***

(4) xmlns:wsman="gost.e-ecolog.ru ***>***

(5) <s:Header>

(6) ...

(7) <wsa:To>транспортный адрес службы</wsa:To>

(8) <wsman:ResourceURI> ResourceURI </wsman:ResourceURI>

(9) <wsman:SelectorSet>

(10) <wsman:Selector Name="имя">значение</wsman:Selector> +

(11) </wsman:SelectorSet> ?

(12) ...

(13) </s:Header>

(14) <s:Body> ... </s:Body>

(15) </s:Envelope>

Следующие определения накладывают дополнительные нормативные ограничения на схему сообщения, представленную выше:

wsman:SelectorSet

оболочка для одного или более элементов Selector, которые требуются для ссылки на экземпляр

wsman:SelectorSet/wsman:Selector

используется для описания селектора и его значения.

Если требуется более одного селектора, то для каждой части полного селектора существует свой элемент Selector. Значение этого элемента является значением Selector.

wsman:SelectorSet/wsman:Selector/@Name

имя селектора (должно обрабатываться нечувствительно к регистру)

Значение селектора может быть вложенная ссылка оконечной точки(EPR).

Пример:

В следующем примере селектор в строке 9 является частью SelectorSet, который содержит вложенную EPR (строки 10—18) со своим собственным адресом, ResourceURI и элементами SelectorSet:

(1) <s:Envelope

(2) xmlns:s="gost.e-ecolog.ru

(3) xmlns:wsa="gost.e-ecolog.ru

(4) xmlns:wsman="gost.e-ecolog.ru ***>***

(5) <s:Header>

(6) ...

(7) <wsman:SelectorSet>

(8) <wsman:Selector Name="Primary"> 123 </wsman:Selector>

(9) <wsman:Selector Name="EPR">

(10) <wsa:EndpointReference>

(11) <wsa:Address>адрес</wsa:Address>

(12) <wsa:ReferenceParameters>

(13) <wsman:ResourceURI> URI ресурса</wsman:ResourceURI>

(14) <wsman:SelectorSet>

(15) <wsman:Selector Name="имя">значение</wsman:Selector>

(16)</wsman:SelectorSet>

(17) </wsa:ReferenceParameters>

(18) </wsa:EndpointReference>

(19) </wsman:Selector>

(20) </wsman:SelectorSet>

(21) ...

(22) </s:Header>

(23) <s:Body> ... </s:Body>

(24) </s:Envelope>

35

R5.4.2.2-5 Для служб, использующих модель адресации WS-Management по умолчанию, значение wsman:Selector должно быть одним из следующих значений:

- простой тип, как определено в пространстве имен XML-схемы

gost.e-ecolog.ru :001/XMLSchema

- вложенный wsa:EndpointReference, использующий модель адресации WS-Management по умолчанию.

Служба может вернуть сообщение об ошибке wsman:InvalidSelectors, если селектор не простого типа и не является EPR.

R5.4.2.2-6 Служба, соответствующая настоящему стандарту, может отклонить любой селектор или вложенный селектор с вложенным EPR, у которого значение wsa:Address отличается от первичного значения wsa:To и не является URI анонимного адреса.

Основная цель для этого вложенного механизма состоит в том, чтобы допустить ресурсы, которые могут отвечать на вопросы о других ресурсах.

R5.4.2.2-7 Служба может не обработать имя селектора размером более 2048 знаков.

R5.4.2.2-8 Служба может не обработать значение селектора размером более 4096 знаков, включая любые вложенные селекторы, и может не обработать сообщение, содержащее более 8096 знаков в корневом элементе SelectorSet.

5.4.2.3 Отказы для модели адресации по умолчанию

Когда генерируются отказы, связанные с информацией в модели адресации, основанной на формате по умолчанию, они могут содержать конкретные коды детали отказа. Данные коды детали перечислены отдельно в 14.6 и не применяются при использовании методов адресации, специфичных для служб.

5.4.3 Ссылки оконечной точки, специфичные для служб

Несмотря на то, что WS-Management определяет модель адресации по умолчанию, в некоторых случаях эта модель не будет доступной или подходящей.

R5.4.3-1 Служба, соответствующая настоящему стандарту может не понять значение заголовка, используемого моделью адресации WS-Management по умолчанию. Если это произошло и если клиент включил в элемент wsman:ResourceURI атрибут mustUnderstand=«true», то служба должна вернуть ошибку s:NotUnderstood.

R5.4.3-2 Служба, соответствующая настоящему стандарту, может потребовать дополнительные заголовочные значения, которые выходят за рамки настоящего стандарта.

Таким образом, службы могут использовать альтернативные модели адресации для ссылки на ресурсы WS-Management. Эти модели адресации могут использовать или не использовать элементы ResourceURI или SelectorSet и, тем не менее, служить допустимыми моделями адресации, если они удовлетворяют правилам адресации.

В дополнение к определенной альтернативной модели адресации служба вообще может не определять явную модель адресации и вместо этого использовать непрозрачный EPR, сгенерированный во время выполнения, который обрабатывается согласно стандартным правилам адресации.

При использовании подобных моделей адресации клиентское приложение должно понимать и взаимодействовать с методами обнаружения для получения EPR, которые выходят за рамки настоящего стандарта.

5.4.4 mustUnderstand

Данный пункт описывает использование атрибута mustUnderstand независимо от того, использует ли реализация адресацию WS-Management (см. 5.1) или адресацию из рекомендации W3CWS-Addressing.

Атрибут mustUnderstand для заголовков SOAPинтерпретируется в WS-Management как инструкция «должен соответствовать». Например, если заголовок SOAP, который описан как необязательный в настоящем стандарте, помечен mustUnderstand= «true», служба должна соответствовать спецификации заголовка или вернуть ошибку. Атрибут mustUnderstand может быть опущен, чтобы гарантировать, что служба рассматривает заголовок как необязательный.

Если wsa:Action URI не понят, реализация, возможно, не знает, как обработать конкретное сообщение. Так, на практике для следующих элементов, отсутствие или наличие mustUnderstand = «true» не имеет никакого реального эффекта на сообщение, так как mustUnderstand неявно подразумевается:

- wsa:To
- wsa:MessageID
- wsa:RelatesTo
- wsa:Action
- wsa:ReplyTo
- wsa:FaultTo

R5.4.4-1 Служба, соответствующая настоящему стандарту, должна обработать любой из предыдущих элементов одинаково вне зависимости от того, присутствует ли mustUnderstand= «true».

Как следствие, клиенты могут опустить mustUnderstand = «true» в любом из указанных выше элементе без изменения в значении.

R5.4.4-2 Если служба не может соответствующим образом обработать заголовок с атрибутом mustUnderstand= «true», то служба должна вернуть сообщение об ошибке s:NotUnderstood.

Цель — терпимость со стороны служб по отношению к непоследовательному использованию клиентами mustUnderstand, чтобы запрос не был истолкован неверно.

Важно, чтобы клиенты, использующие модель адресации WS-Management по умолчанию (ResourceURI и SelectorSet), использовали mustUnderstand= «true» в элементе wsman:ResourceURI, чтобы гарантировать совместимость службы с этой моделью адресации. В ином случае, реализации, использующие специфичные для служб модели адресации, будут потенциально игнорировать эти значения заголовка и вести себя несовместимо с намерениями клиента.

5.4.5 wsa:To

Данный пункт описывает использование заголовка адресации wsa:To независимо от того, использует реализация адресацию WS-Management (см. 5.1) или версию рекомендации W3CWS-Addressing.

В сообщениях-запросах адрес wsa:To содержит транспортный адрес службы. В некоторых случаях данного адреса достаточно для определения местонахождения ресурса. В других случаях служба обрабатывает сообщения для ресурсов с множеством экземпляров. В этих случаях сообщение, как правило, содержит дополнительные заголовки, чтобы служба могла определить ресурс в рамках своей области. Например, когда используется модель адресации по умолчанию, эти дополнительные заголовки будут элементами ResourceURI и SelectorSet.

Примечание — WS-Management не запрещает сосуществование нескольких служб-слушателей на одной физической платформе. Такие службы можно обнаружить и различить, используя механизмы, определенные за рамками настоящего стандарта.

R5.4.5-1 Заголовок wsa:To должен присутствовать во всех сообщениях, будь то запросы, ответы или события. При отсутствии иных требований рекомендуется, чтобы сетевой адрес для ресурсов, требующих аутентификацию, заканчивался последовательностью символов */wsman*. Если используется */wsman*, неаутентифицированный доступ не разрешен.

(1) <wsa:To>hgost.e-ecolog.ru

R5.4.5-2 При отсутствии других требований рекомендуется, чтобы сетевой адрес для ресурсов, не требующих аутентификацию, заканчивался последовательностью символов */wsman-anon*. Если используется */wsman-anon*, то аутентифицироавнный доступ не требуется.

(1) <wsa:To>hgost.e-ecolog.ru

Включение сетевого транспортного адреса в сообщение SOAP может казаться избыточным, поскольку сетевое соединение уже установлено клиентом. Однако в случаях, когда сообщение проходит через узлы-посредники, требуется сетевой транспортный адрес, чтобы узлы-посредники могли исследовать сообщение и установить связь с фактической оконечной точкой.

Заголовок wsa:To может включать любое число символов, требуемых для определения местонахождения службы и группы ресурсов в рамках данной службы.

R5.4.5-3 Служба должна сгенерировать ошибку, когда адрес wsa:To не может быть обработан в следующих ситуациях:

- если ресурс не в сети, возвращается ошибка wsa:EndpointUnavailable со следующим кодом детализации:

gost.e-ecolog.ru

- если ресурс не может быть установлен («не найден»), возвращается ошибка wsa:DestinationUnreachable.
- если ресурс действующий, но происходят внутренние ошибки, возвращается ошибка wsman:InternalError.
- если к ресурсу нельзя получить доступ из-за соображений безопасности, возвращается ошибка wsman:AccessDenied.

5.4.6 Другие заголовки адресации

Данный пункт описывает использование других заголовков адресации, независимо от того, использует ли реализация адресацию WS-Management (см. 5.1) или версию рекомендации W3C WS-Addressing.

WS-Management зависит от спецификации адресации для описания правил для использования других заголовков адресации.

5.4.6.1 Обработка заголовков адресации

Следующие дополнительные блоки заголовков, связанные с адресацией, встречаются в сообщениях WS-Management.

R5.4.6.1-1 Служба, соответствующая настоящему стандарту, должна распознавать и обрабатывать следующие блоки заголовков адресации:

- **wsa:To;**
- **wsa:ReplyTo** (обязательный, когда ожидается ответ);
- **wsa:FaultTo** (необязательный);
- **wsa:MessageID** (обязательный);
- **wsa:Action** (обязательный);
- **wsa:RelatesTo** (обязательный в ответах).

Использование этих блоков заголовка обсуждается в последующих параграфах.

5.4.6.2 wsa:ReplyTo

WS-Management требует соблюдать в адресации wsa:ReplyTo следующие правила:

R5.4.6.2-1 Заголовок wsa:ReplyTo должен присутствовать во всех сообщениях-запросах, когда требуется ответ. Данный адрес должен быть либо действительным адресом для нового соединения, использующего любой поддерживаемый службами транспорт, либо URI анонимного адреса, который указывает, что ответ должен передаваться по тому же соединению, по которому прибыл запрос. Если заголовок wsa:ReplyTo отсутствует, возвращается ошибка wsa:MessageInformationHeader Required.

Некоторые сообщения, такие как доставки событий, SubscriptionEnd и так далее, не требуют ответа и могут опустить элемент wsa:ReplyTo.

R5.4.6.2-2 Служба, соответствующая настоящему стандарту, может потребовать передачу всех ответов по тому же соединению, по которому поступает запрос. В таком случае на это должен указывать URI, как описано в R5.4.6.2-1. В противном случае служба должна возвратить ошибку wsman:UnsupportedFeature со следующим кодом детализации: gost.e-ecolog.ru
wsman/faultDetail/ AddressingMode

R5.4.6.2-3 При доставке событий, для которых требуется подтверждение доставки, отправитель события должен включать элемент wsa:ReplyTo и следовать требованиям 10.8.

R5.4.6.2-4 Это правило преднамеренно оставлено незаполненным.

R5.4.6.2-5 Это правило преднамеренно оставлено незаполненным.

Адресация позволяет клиентам включать свои специфические параметры ссылки в заголовки wsa:ReplyTo. Адресация требует, чтобы эти параметры ссылки были извлечены из запросов, помещены в ответы с удалением обертки ReferenceParameters и размещены как заголовки SOAP верхнего уровня в ответе (см. 5.1). Это позволяет клиентам лучше соотносить ответы с исходными запросами. Данный шаг не может быть опущен.

Пример:

В следующем примере заголовок x:someHeader включен в ответное сообщение:

```
(1)  <s:Envelope
(2)  xmlns:s="gost.e-ecolog.ru                    03/05/soap-envelope"
(3)  xmlns:wsa="gost.e-ecolog.ru
(4)  xmlns:wsman="gost.e-ecolog.ru                                    >
(5)  <s:Header>
(6)   ...
(7)  <wsa:To> gost.e-ecolog.ru                    :/wsa:To>
(8)  <wsa:ReplyTo>
(9)  <wsa:Address>
(10) gost.e-ecolog.ru
(11) </wsa:Address>
(12) <wsa:ReferenceParameters>
(13)  <x:someHeaderxmlns:x="...">содержимое пользователя</x:someHeader>
(14) </wsa:ReferenceParameters>
(15) </wsa:ReplyTo>
(16)  ...
(17) </s:Header>
(18) <s:Body> ... </s:Body>
(19) </s:Envelope>
```

R5.4.6.2-6 Если адрес wsa:ReplyTo неприменим или отсутствует, службе следует не отвечать на запрос и закрыть или завершить соединение согласно правилам нижележащего сетевого транспортного протокола. В таких случаях службе следует зарегистрировать в журнале некоторый тип записи, чтобы позже помочь диагностировать дефект клиента.

5.4.6.3 wsa:FaultTo

Данный пункт представляет требования WS-Management к использованию wsa:FaultTo.

R5.4.6.3-1 Служба, соответствующая настоящему стандарту, может поддерживать адрес wsa:FaultTo, отличающийся от адреса wsa:ReplyTo. Если служба не поддерживает такие запросы, то должен быть возвращен отказ wsman:UnsupportedFeature со следующим кодом детализации: gost.e-ecolog.ru

Если в запросе опущены заголовки wsa:FaultTo и wsa:ReplyTo, то обычно используются механизмы транспортного уровня, чтобы послать сообщения об ошибке в запросе, потому что адрес, по которому нужно отправить отказ, неопределен. В данном случае не будет ошибкой службе просто разорвать соединение.

R5.4.6.3-2 Если wsa:FaultTo опущен, то служба должна вернуть отказ по адресу wsa:ReplyTo, если таковой отказ произойдет.

R5.4.6.3-3 Служба, соответствующая настоящему стандарту, может потребовать, чтобы все отказы были поставлены клиенту по тому же самому транспорту или соединению, по которому поступает запрос. В этом случае URI должен быть URI анонимного адреса. Если служба не поддерживает отдельно адресуемую доставку отказа и wsa:FaultTo содержит любой другой адрес, то должен быть возвращен отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Примечание — Данная спецификация не запрещает более полным реализациям полностью поддерживать wsa:FaultTo.

R5.4.6.3-4 Это правило преднамеренно оставлено незаполненным.

Пример:

В следующем примере заголовок x:someHeader включен в сообщения отказа, если таковые происходят.

```
(1)  <s:Envelope
(2)  xmlns:s="gost.e-ecolog.ru
(3)  xmlns:wsa="gost.e-ecolog.ru
(4)  xmlns:wsman="gost.e-ecolog.ru                                    >
(5)  <s:Header>
(6)  ...
(7)  <wsa:To> gost.e-ecolog.ru                  :/wsa:To>
(8)  <wsa:FaultTo>
(9)  <wsa:Address>
(10) gost.e-ecolog.ru
(11) </wsa:Address>
(12) <wsa:ReferenceParameters>
(13) <x:someHeaderxmlns:x="...">содержимое пользователя</x:someHeader>
(14) </wsa:ReferenceParameters>
(15) </wsa:FaultTo>
(16) ...
(17) </s:Header>
(18) <s:Body> ... </s:Body>
(19) </s:Envelope>
```

R5.4.6.3-5 Если адрес wsa:FaultTo не применим, службе следует не отвечать на запрос. Подобным образом, если нет подходящего адреса, соответствующего правилам WS-Adressing, для отправки отказа, службе следует не отвечать и закрыть сетевое соединение. В таких случаях службе следует зарегистрировать в журнале некоторый тип записи, чтобы позже помочь диагностировать дефект клиента.

R5.4.6.3-6 Служба должна соответствующим образом дублировать элемент wsa:Address из элемента wsa:FaultTo в элемент wsa:To ответа, даже если часть информации не будет понята службой.

Данное правило применяется в случаях, когда клиент включает частные суффиксы содержания в адрес HTTP или HTTPS, которые служба не понимает. Если служба удаляет эту информацию, строя адрес, получившееся сообщение может обработаться некорректно.

5.4.6.4 wsa:MessageID и wsa:RelatesTo

WS-Management определяет использование wsa:MessageID и wsa:RelatesTo следующим образом:

R5.4.6.4-1 URI для MessageID и RelatesTo могут иметь любой формат при условии, что они являются корректными URI согласно RFC 3986. Два URI считаются различными, даже если символы в этих URI отличаются только регистром.

Настоящий стандарт рекомендует следующие два формата. Оптимальным считается первый, так как он поддерживается RFC 4122:

urn:uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx или

uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

В этих форматах каждый *x* представляет собой заглавную или строчную шестнадцатеричную цифру (строчные буквы требуются по RFC 4122); нет пробелов или иных символов. Значение может быть универсальным уникальным идентификатором (UUID) в стиле DCE с доказуемыми свойствами уникальности в данном формате, однако, нет необходимости иметь доказуемые свойства уникальности в URI, используемых в заголовках wsa:MessageID и wsa:RelatesTo.

Независимо от формата URI не должны превышать максимум, определенный в R13.1-6.

UUID имеют числовое значение, а также строковое значение, что может привести к неоднозначности. UUID URI, записанный строчными буквами отличается от того же самого UUID в верхнем регистре. Это вызвано тем, что URI чувствительны к регистру. Если UUID преобразован в десятичный эквивалент, исходный регистр символов потерян. WS-Management работает с самими значениями URI, а не с лежащим в основе эквивалентным десятичным представлением. Службы могут свободно интерпретировать URI любым способом, но им не разрешается менять используемый регистр при повторе сообщения или любого из значений MessageID в последующих сообщениях.

RFC 4122 требует, чтобы цифры были представлены строчными буквами, что является ответственностью клиента. Служба просто обрабатывает значения как значения URI и от нее не требуется анализировать URI на предмет правильности или соответствия. Служба копирует то, что использовал клиент, в ответном заголовке wsa:RelatesTo, и ей не разрешается изменять использованный регистр.

R5.4.6.4-2 Следует генерировать MessageID в соответствии с любым алгоритмом, который гарантирует, что никакие два MessageIDs не повторяются. Поскольку значение рассматривается как чувствительное к регистру (R5.4.6.4-1), может возникнуть ошибка обработки, если то же самое значение используется снова, отличаясь только регистром. В результате служба не должна создавать или использовать значения MessageID, которые отличаются только регистром. Для любого сообщения, переданного службой, MessageID не должен повторно использоваться.

Клиент гарантирует, что значение MessageID не используется в запросах повторно. Хотя службы и клиенты могут выпускать различные MessageIDs, которые отличаются только регистром, служба не обязана обнаруживать это различие, при этом не требуется анализировать URI на предмет синтаксической правильности или повторного использования.

R5.4.6.4-3 Элемент RelatesTo должен присутствовать во всех ответных сообщениях и отказах, содержать MessageID связанного сообщения-запроса, и регистр должен соответствовать оригиналу и обрабатываться как значение URI, а не как бинарное значение UUID.

R5.4.6.4-4 Если MessageID невозможно разобрать синтаксически или отсутствует, следует вернуть отказ wsa:InvalidMessageInformationHeader.

Пример:

Следующие примеры иллюстрируют использование wsa:MessageID:

(20) <wsa:MessageID>

(21) uuid:d9726315-bc91-430b-9ed8-ce5ffb858a91

(22) </wsa:MessageID>

(23)

(24) <wsa:MessageID>

(25) anotherScheme:ID/12310/1231/16607/25

(26) </wsa:MessageID>

5.4.6.5 wsa:Action

wsa:Action URI указывает на «операцию», которая выполняется над ресурсом.

R5.4.6.5-1: wsa:Action URI не должны использоваться для определения конкретного класса ресурса или экземпляра, они только определяют операцию для выполнения над этим ресурсом.

R5.4.6.5-2: Для всех оконечных точек ресурса служба должна вернуть отказ wsa:ActionNotSupported, если требуемое действие не поддерживается службами для указанного ресурса.

Другими словами, чтобы моделировать «Get» элемента «Disk», wsa:Action URI содержит «Get». Заголовок wsa:To и потенциально другие заголовки SOAP указывают, к чему именно осуществляется доступ. Например, когда используется модель адресации по умолчанию, ResourceURI, как правило, содержит ссылку на «Disk», и SelectorSet определяет конкретный диск. Другие модели адресации, специфичные для служб, могут задавать идентичность ресурса иными способами.

Реализации могут свободно поддерживать дополнительные нестандартные методы, которые объединяют понятия «Get» и «Disk» в единственное действие «GetDisk», если они стремятся поддерживать отделенную форму для максимизации функциональной совместимости. Один из основных моментов, лежащих в основе WS-Management,— это унификация общепринятых методов по мере возможности.

R5.4.6.5-3 Если служба демонстрирует какой-либо из следующих типов возможностей, то служба, соответствующая настоящему стандарту, должна, по крайней мере, предоставить возможность, используя определения в таблице 4 согласно правилам настоящего стандарта. Служба может также предоставить дополнительную схожую функциональность, используя отдельный URI wsa:Action.

Таблица 4 — Описания wsa:Action URI

URI действия	Описание
gost.e-ecolog.ru	Моделирует любое простое получение единичного объекта
gost.e-ecolog.ru	Ответ на «Get»
gost.e-ecolog.ru	Моделирует обновление всего объекта
gost.e-ecolog.ru	Ответ на «Put»
gost.e-ecolog.ru	Моделирует создание нового объекта
gost.e-ecolog.ru	Ответ на «Create»
gost.e-ecolog.ru	Моделирует удаление объекта
gost.e-ecolog.ru	Ответ на «Delete»
gost.e-ecolog.ru	Начинает перечисление или запрос
gost.e-ecolog.ru	Ответ на «Enumerate»
gost.e-ecolog.ru	Получает следующий набор результатов перечисления
gost.e-ecolog.ru	Ответ на «Pull»
gost.e-ecolog.ru	Возобновляет действие счетчика, который, возможно, завершился (не требуется в WS-Management)
gost.e-ecolog.ru	Ответ на «Renew» (не требуется в WS-Management)
gost.e-ecolog.ru	Получает статус перечисления (не требуется в WS-Management)
gost.e-ecolog.ru	Ответ на «GetStatus» (не требуется в WS-Management)
gost.e-ecolog.ru	Прекращает активное перечисление
gost.e-ecolog.ru	Ответ на «Release»
gost.e-ecolog.ru	Уведомляет, что перечисление завершено (не требуется в WS-Management)
gost.e-ecolog.ru	Моделирует подписку на источник события
gost.e-ecolog.ru	Ответ на «Subscribe»
gost.e-ecolog.ru	Возобновляет подписку до ее истечения
gost.e-ecolog.ru	Ответ на «Renew»
gost.e-ecolog.ru	Запрашивает статус подписки
gost.e-ecolog.ru	Ответ на «GetStatus»
gost.e-ecolog.ru	Удаляет активную подписку
gost.e-ecolog.ru	Ответ на «Unsubscribe»

Окончание таблицы 4

URI действия	Описание
gost.e-ecolog.ru	Передает сообщение, чтобы указать, что подписка закончилась
gost.e-ecolog.ru	Поставляет набор событий, основанный на подписке
gost.e-ecolog.ru	Псевдособытие, которое моделирует пульсацию активной подписки; отправляется, когда реальные события отсутствуют, но используется для того, чтобы указывать, что подписка событий и механизм доставки тем не менее активны
gost.e-ecolog.ru	Псевдособытие, которое указывает, что реальное событие было пропущено
gost.e-ecolog.ru	Используется подписчиками событий, чтобы подтвердить получение событий; позволяет строго упорядочить потоки событий
gost.e-ecolog.ru	Используется для единичного события, которое не определяет его собственное действие

R5.4.6.5-4 Специализированное действие может быть поддержано, если операцией является специализированный метод, семантическое значение которого не присутствует в таблице 4.

R5.4.6.5-5 Все уведомления должны содержать уникальный URI действия, который определяет тип доставки событий. Для единичных уведомлений только с одним событием в сообщении (режим доставки gost.e-ecolog.ru), wsa:Action URI определяет тип события. Для других режимов доставки действие варьируется, как описано в 10.2.7.

5.4.6.6 wsa:From

Заголовок wsa:From может использоваться в любых сообщениях, ответах или событиях, для указания на источник. Когда соединение используется для запроса и для ответа, данный заголовок не предоставляет полезной информации, но может быть полезным в случаях, когда ответ прибывает по другому соединению.

R5.4.6.6-1 Служба, соответствующая настоящему стандарту, может включать в сообщение адрес wsa:From. Службе, соответствующей настоящему стандарту, следует обрабатывать любое входящее сообщение с элементом wsa:From.

R5.4.6.6-2 Совместимой службе не следует отвергать сообщение с элементом wsa:From вне зависимости от того, включен ли атрибут mustUnderstand.

Примечание — Обработка wsa:From заголовка тривиальна, так как она не имеет никакого влияния на значение сообщения. Адрес From прежде всего нужен для целей аудита и ведения журнала событий.

6 Управляющие заголовки WS-Management

WS-Management определяет несколько заголовков SOAP, которые могут использоваться с любой операцией.

6.1 wsman:OperationTimeout

Большинство управляющих операций строго ограничены во времени из-за ограничений качества обслуживания и обязательств. Если операции не могут быть закончены в требуемое время, служба возвращает отказ, чтобы клиент мог выполнить свои обязательства. Следующее значение заголовка может поставляться любым сообщением WS-Management, чтобы указать, что клиент ожидает ответ или отказ в течение требуемого времени:

(1) <wsman:OperationTimeout>xs:duration</wsman:OperationTimeout>

R6.1-1 Все сообщения-запросы могут содержать элемент заголовка wsman:OperationTimeout, который указывает на максимальное количество времени, в течение которого клиент готов ждать отправки ответа службы. Службе следует интерпретировать обратный отсчет времени как начинающийся с момента, когда обработано сообщение, до момента, когда будет произведен ответ.

R6.1-2 Если время обратного отсчета превышено, а операция еще не завершена, службе следует немедленно отправить отказ wsman:TimedOut. Если значение OperationTimeout недействительно, следует возвратить отказ wsa:InvalidMessageInformationHeader.

R6.1-3 Если служба не поддерживает функцию времени ожидания, заданного пользователем, то следует вернуть отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

R6.1-4 Если элемент wsman:OperationTimeout опущен, служба может интерпретировать отсутствие wsman:OperationTimeout как инструкцию заблокироваться на неопределенное время до получения ответа либо использовать значение времени ожидания по умолчанию.

Эти правила не запрещают службе поддерживать бесконечные или очень длинные времена ожидания. Поскольку сетевые соединения редко блокируются на неопределенное время без передачи сообщений, есть вероятность истечения времени ожидания транспортного уровня. Также обратный отсчет начинается с момента, когда получено сообщение; таким образом, сетевая задержка не учитывается. Если клиенту нужно определить диапазон фактических времен ожидания или значений по умолчанию, он может запросить метаданные, но формат таких метаданных выходит за рамки настоящего стандарта.

Если время ожидания истекает таким образом, что служба уже выполнила часть работы, связанной с запросом, служба переходит в аномальное состояние. Данная спецификация не рассматривает поведение в такой ситуации. Очевидно, что служба может попытаться отменить эффекты любых частично выполненных операций, но это не всегда практично. В таких случаях служба может вести внутренний журнал запросов и операций, который клиент позднее может запросить.

Например, если при операции Delete истекает время ожидания, служба решает восстановить удаленные файлы или продолжить удаление даже при том, что отправлен отказ wsman:TimedOut. Служба может включить в отказ дополнительную информацию (см. 14.5), описывающую внутреннюю политику в этом отношении. Служба может попытаться вернуться в состояние, которое существовало до начала операции, но это не всегда возможно.

R6.1-5 Если атрибут mustUnderstand применен к элементу wsman:OperationTimeout и служба понимает wsman:OperationTimeout, то служба должна принять требуемое значение или вернуть сообщение-отказ в соответствии с R6.1-2. Службе следует попытаться завершить запрос в течение требуемого времени или выпустить отказ без дальнейшей задержки.

Клиенты всегда могут опускать заголовок mustUnderstand для унификации поведения всех реализаций. Служба, соответствующая настоящему стандарту, может игнорировать значение времени ожидания или рассматривать его как рекомендацию, если отсутствует mustUnderstand.

Пример:

Ниже приведен пример правильно заданного 30-секундного ожидания в заголовке SOAP:

(1) <wsman:OperationTimeout>PT30S</wsman:OperationTimeout>

Если время ожидания на транспортном уровне истекает до фактического wsman:OperationTimeout, то операцию можно рассматривать (как указано в 13.3) как ошибку соединения. На практике время ожидания сетевого транспорта может быть задано более длительным, чем любой ожидаемый wsman:OperationTimeout.

6.2 wsman:MaxEnvelopeSize

Чтобы предотвратить ситуацию, когда ответ превосходит возможности клиента, сообщение-запрос может содержать ограничение на размер ответа.

Следующее значение заголовка может поставляться любым сообщением WS-Management для указания на то, что клиент ожидает ответ, полный конверт SOAP которого не должен превышать конкретное число октетов:

(1) <wsman:MaxEnvelopeSize> xs:positiveInteger</wsman:MaxEnvelopeSize>

Ограничение накладывается на весь конверт. Для реализаций с ограниченными ресурсами нужны надежные значения, чтобы предоставить необходимое число памяти для всей обработки SOAP, а не только элемента Body SOAP.

R6.2-1 Все сообщения-запросы могут содержать элемент заголовка wsman:MaxEnvelopeSize, который указывает на максимальное число октетов (не символов) в ответе во всем конверте SOAP. Если служба не может составить ответ в пределах требуемого размера, следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

R6.2-2 Если атрибут mustUnderstand установлен в «true», то служба должна соблюсти ограничение. Если ответ превысит максимальный размер, следует вернуть сообщение-отказ wsman:EncodingLimit. Поскольку служба может выполнять операцию, не зная заранее размер ответа, служба должна отменить любые эффекты операции, прежде чем отправить отказ. Если операция не может быть полностью инвертирована (например, разрушающие Put или Delete, или Create), служба должна указать в отказе wsman:EncodingLimit, что операция успешно завершена, следующим кодом детализации:

gost.e-ecolog.ru

R6.2-3 Если атрибут mustUnderstand установлен в «false», служба может проигнорировать заголовок.

R6.2-4 Службам следует отклонять любое значение MaxEnvelopeSize меньшее, чем 8192 октета. Это число — безопасный минимум, в котором отказы могут быть достоверно закодированы для всех наборов символов. Если требуемый размер меньше, чем указанный лимит, следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

У службы может быть свой собственный предел кодирования, независимый от того, что определяет клиент, и в этом случае применяется такой же отказ.

R6.2-5 Если служба не может составить ответ в пределах своих собственных внутренних пределов, следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

Определение в схеме элемента wsman:MaxEnvelopeSize содержит атрибут Policy, поскольку данный элемент используется также для других целей. Данная спецификация не определяет использование атрибута Policy, когда элемент wsman:MaxEnvelopeSize используется в качестве заголовка SOAP.

R6.2-6 Клиентам не следует добавлять атрибут Policyв элемент wsman:MaxEnvelopeSize, когда он используется в качестве заголовка SOAP. Службам следует игнорировать атрибут Policy, если он появляется в элементе wsman:MaxEnvelopeSize, когда тот используется в качестве заголовка SOAP.

6.3 wsman:Locale

Операции управления часто связаны с языковыми параметрами, при этом в ответах может потребоваться перевод многих объектов. Как правило, перевод требуется для предназначенной для читателей описательной информации, возвращаемой в ответе. Если клиент требует, чтобы такие результаты были переведены на определенный язык, он может использовать необязательный заголовок wsman:Locale, который использует стандартный атрибут XMLxml:lang, следующим образом:

(1) <wsman:Locale xml:lang = “xs:language” s:mustUnderstand = “false” />

R6.3-1 Если атрибут mustUnderstand опущен или установлен как «false», службе следует использовать это значение при формировании ответного сообщения и адаптировать любые локализуемые значения соответствующим образом. Такое использование рекомендуется для большинства случаев. Языковой параметр в этом случае рассматривается как рекомендация.

R6.3-2 Если атрибут mustUnderstand установлен как «true», служба должна гарантировать, что ответы содержат локализованную информацию во всех соответствующих местах, или иначе служба должна отправить отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Служба может всегда возвращать отказ, если wsman:Locale содержит s:mustUnderstand, установленный равным «true», так как, возможно, она не сможет гарантировать, что ответ локализован.

Некоторые реализации делегируют запрос к другой подсистеме для обработки, таким образом, служба не может быть уверена, что локализация фактически произошла.

R6.3-3 Значение атрибута xml:lang в заголовке wsman:Locale должно быть действительным языковым кодом RFC 5646.

R6.3-4 При любом ответе, событии или единичном сообщении службе следует включать атрибут xml:lang в элемент s:Envelope (или другие элементы), чтобы сигнализировать приемнику, что в теле сообщения появляется локализованное содержимое. Данный атрибут может быть опущен, если в теле отсутствует описательное содержимое. Включение атрибута xml:lang не является ошибкой, даже если описательное содержимое отсутствует.

Пример:

(1) <s:Envelope

(2) xml:lang="en-us"

(3) xmlns:s="gost.e-ecolog.ru

(4) xmlns:wsa="gost.e-ecolog.ru

(5) xmlns:wsman="gost.e-ecolog.ru >

(6) <s:Header> ... </s:Header>

(7) <s:Body> ... </s:Body>

(8) </s:Envelope>

Атрибут xml:lang может появиться в сообщении в любом содержимом, хотя более простой подход позволяет клиенту всегда проверять наличие атрибута в одном месте — в обертке s:Envelope.

R6.3-5 Для операций, которые охватывают последовательности нескольких сообщений, элемент wsman:Locale обрабатывается только в первоначальном сообщении, его следует игнорировать в последующих сообщениях, так как первое сообщение устанавливает необходимый языковой параметр. Служба может вернуть сообщение-отказ, если в последующих сообщениях присутствует wsman:Locale и его значение отличается от используемого в первоначальном запросе.

Это правило применяется прежде всего для сообщений Enumerate и Pull. Языковой параметр ясным образом установлен во время первоначального запроса Enumerate, поэтому изменение языкового параметра во время перечисления бессмысленно. Служба игнорирует любые элементы wsman:Locale в последующих сообщениях Pull, однако клиент может убедиться, что между запросами Pull значение не меняется. Эта унификация упрощает клиенту построение сообщений.

Рекомендуется (как установлено в R6.3-1), чтобы элемент wsman:Locale никогда не содержал атрибут mustUnderstand. Таким образом, клиент не получит отказы в неожиданных местах.

6.4 wsman:OptionSet

Заголовок OptionSet используется для передачи службе набора переключателей, позволяющих изменить запрос или уточнить его характер. Это средство предназначено для того, чтобы помочь службе наблюдать желаемые для клиента контекст или сторонние эффекты, не меняя схемы результата или значения адресации. Опции подобны ключам, используемым в оболочках командной строки, в том смысле, что они являются текстовыми расширениями, специфичными для службы.

R6.4-1 Любое сообщение-запрос может содержать заголовок wsman:OptionSet, который оборачивает ряд дополнительных параметров. Эти параметры помогают службе составить желаемый ответ или наблюдать требуемый побочный эффект.

R6.4-2 Службе не следует отправлять ответы, неподтвержденные события или единичные сообщения, содержащие заголовки wsman:OptionSet, если она не выступает в роли клиента по отношению к другой службе. Такие заголовки предназначены для сообщений-запросов, по которым ожидается последующий ответ, включая события с подтверждением.

R6.4-3 Если в блоке OptionSet атрибут mustUnderstand опущен или если он присутствует со значением «false», служба может проигнорировать весь блок wsman:OptionSet. Если он присутствует с значением «true» и при этом служба не поддерживает wsman:OptionSet, то служба должна вернуть сообщение-отказ s:NotUnderstood.

Службы могут обработать блок OptionSet, если таковой присутствует, но они не обязаны понимать или обрабатывать отдельные опции, как показано в R6.4-6. Однако, если MustComply установлен в «true» в какой-либо опции, mustUnderstand необходимо тоже установить как «true». Таким образом можно избежать несовместимости при разрешении проигнорировать весь блок OptionSet, в то же время имея в отдельных опциях MustComply.

R6.4-4 Каждый класс ресурса может поддерживать свой собственный набор опций, более того отдельный экземпляр класса ресурса может поддерживать свой собственный набор опций. Не требуется, чтобы опции одинаково использовались в различных классах и экземплярах ресурсов. Форматы

метаданных и определения опций выходят за рамки настоящего стандарта и могут быть различными для различных служб.

R6.4-5 Любое число отдельных элементов опций может появиться внутри элемента wsman:OptionSet. При необходимости имена опций могут повторяться. Содержимое должно задаваться простой строкой (xs:string). Данная спецификация не устанавливает ограничения для имен и значений по чувствительности/нечувствительности к регистру. Однако использованный регистр должен сохраняться при передаче сообщения, содержащего элемент OptionSet, и его содержания, через узлы-посредники SOAP.

Интерпретация чувствительности опции к регистру зависит от службы и определения конкретной опции, так как значение может проходить через подсистемы реального мира, которые могут оказаться несовместимыми в обработке регистра. Клиент, заинтересованный в функциональной совместимости, может опустить атрибуты mustUnderstand и MustComply.

R6.4-6 Отдельные значения опций могут быть рекомендациями или могут запрашиваться клиентом. Служба должна выполнять все опции, отмеченные атрибутом MustComply, установленным равным «true», или вернуть сообщение об ошибке wsman:InvalidOptions со следующим кодом детализации:

gost.e-ecolog.ru ɔrg/wbem/wsman/1/wsman/faultDetail/NotSupported

Любая опция, не отмеченная этим атрибутом (либо, если атрибут установлен равным «false»), является рекомендованной для службы, и служба может проигнорировать ее. Если какая-либо опция отмечена атрибутом MustComply, установленным равным «true», то атрибут mustUnderstand должен использоваться на всем блоке wsman:OptionSet.

Эта возможность требуется, когда служба делегирует интерпретацию и выполнение вариантов другому компоненту. Во многих случаях процессор SOAP не может знать, поддерживается ли опция, и может только передать его следующей подсистеме.

R6.4-7 Опции могут содержать атрибут Type, который указывает на тип данных содержания элемента Option. Служба может потребовать наличие данного атрибута в любой заданной опции и его установки в QName действительного типа данных схемы XML. Данная версия WS-Management поддерживает только стандартные простые типы, объявленные в пространстве имен gost.e-ecolog.ru 001/XMLSchema.

Это правило может помочь некоторым службам различать числовой тип или типы даты и времени от других строковых значений.

R6.4-8 Опции не следует использовать как заменители документированого метода параметризации сообщений; их следует использовать только как модификаторы.

Опции прежде всего используются для установления контекста, либо в ином случае, инструктируют службу при выполнении операции выполнить побочные операции, такие как включение журнала событий или прослеживания.

R6.4-9 Службе следует вернуть следующие отказы:

- если опции не поддерживаются, **wsman:InvalidOptions** со следующим кодом детализации:

gost.e-ecolog.ru

- когда одно или более имен опции недействительны или не поддерживаются определенным ресурсом, **wsman:InvalidOptions** со следующим кодом детализации:

gost.e-ecolog.ru

- когда значение неправильно для имени опции, **wsman:InvalidOptions** со следующим кодом детализации:

gost.e-ecolog.ru

R6.4-10 Для операций, которые охватывают последовательности нескольких сообщений, элемент wsman:OptionSet обрабатывается только в первоначальном сообщении. Его следует проигнорировать в последующих сообщениях, потому что первое сообщение устанавливает необходимый набор опций. Служба может отправить отказ, если элемент wsman:OptionSet присутствует в последующих сообщениях и его значение отличается от используемого в первоначальном запросе либо в таких сообщениях служба может проигнорировать значение wsman:OptionSet.

Это правило применяется прежде всего для сообщений Enumerate и Pull. Набор опций устанавливается один раз во время первоначального запроса Enumerate, поэтому изменение опций во время перечисления было бы ошибкой.

Опции предназначены для того, чтобы сделать операции более эффективными или предварительно обработать выходные данные от имени клиента. Например, опции могут указать службе, что

возвращаемые значения должны быть вычислены повторно и не должны использоваться кэшированные значения, либо в ответе могут быть опущены любые необязательные значения. Также опции могут использоваться для указания уровня подробностей ответа в рамках схемы XML, связанной с ответом.

Значения опции не предназначены для XML содержимого. Если требуется задать входные данные в XML, то правильной моделью для операции будет создать специализированную операцию со своим собственным wsa:Action URI. Это гарантирует, что в хорошо известные типы сообщений не добавлены скрытые параметры, закладки. Например, при оправке запроса Subscribe, сообщение уже определяет способ задания службе фильтра событий, таким образом, опция не используется для обхода этого и передачи фильтра посредством дополнительного метода.

Пример:

Ниже приведет пример wsman:OptionSet:

(1) <s:Envelope

(2) xmlns:s="gost.e-ecolog.ru

(3) xmlns:wsa="gost.e-ecolog.ru

(4) xmlns:wsman="gost.e-ecolog.ru

(5) xmlns:xs="http://www.w3.org/2001/XMLSchema">

(6) <s:Header>

(7) ...

(8) <wsman:OptionSet s:mustUnderstand="true">

(9) <wsman:Option Name="VerbosityLevel" Type="xs:int">

(10) 3

(11) </wsman:Option>

(12) <wsman:Option Name="LogAllRequests" MustComply ="true"/>

(13) </wsman:OptionSet>

(14) ...

(15) </s:Header>

(16) <s:Body> ... </s:Body>

(17) </s:Envelope>

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения.wsman:OptionSet используется для обертки отдельных блоков опции.

В данном примере s:mustUnderstand установлен как «true», с указанием, что клиент требует, чтобы служба обработала блок опции в соответствии с установленными правилами: wsman:OptionSet/ wsman:Option /@Name идентифицирует опцию (xs:string), может быть простым именем или URI.

Интерпретация имени определяется ресурсом, к которому оно применяется. Имя может повторяться в последующих элементах. Имя не может быть пустым, может быть коротким неконфликтующим URI, определенным поставщиком: wsman:OptionSet/wsman:Option / MustComply если установлено равным «true», указывает, что опция должна быть обработана; в противном случае указывает на оповещение или рекомендацию wsman:OptionSet/wsman:Option /@Type (необязательный), если присутствует, то указывает на тип данных содержимого элемента, который помогает службе интерпретировать содержимое.

Служба может потребовать присутствия данного атрибута на любом заданном элементе опции: wsman:OptionSet/wsman:Option содержимое опции. Значение может быть произвольной простой строкой. Если значение опции пусто, опцию следует интерпретировать как логическое «true», и опция следует «разрешить». Следующий пример разрешает опцию «Verbose»:

(1) <wsman:Optionname= «Verbose»/>.

Опции считаются логически ложными, если они не присутствуют в сообщении. Во всех остальных случаях требуется, чтобы значение опции задавалось явно строкой. Аргументацией в пользу повторения одной и той же опции может быть предоставление спецификации списка опций с тем же самым именем.

6.5 wsman:RequestEPR

Некоторые служебные операции, включая «Put», в состоянии изменять представление ресурса таким образом, что обновление приведет к логическому изменению идентичности ресурса, например «переименование» документа. Во многих случаях это изменение, в свою очередь, меняет EPR данного ресурса после завершения операции, поскольку часто EPR динамически порождаются из значений имени в рамках представления самого ресурса. Это поведение распространено в реализациях SOAP, которые делегируют операции нижележащим системам.

Чтобы предоставить клиенту определить способ, когда такое изменение произошло, определены два заголовка SOAP для запроса и возврата EPR экземпляра ресурса.

В любом сообщении-запросе WS-Management может появиться следующий заголовок:

(1) <wsman:RequestEPR .../>

R6.5-1: Службе, получившей сообщение, которое содержит блок заголовка wsman:RequestEPR, следует вернуть ответ, содержащий блок заголовка wsman:RequestedEPR. Данный блок содержит самый новый EPR ресурса, к которому получен доступ или код статуса, если служба не может определить или возвратить EPR. Данный EPR отражает любые изменения идентичности, которые, возможно, произошли в результате текущей операции, как указано в следующих функциональных возможностях. Блок заголовка в соответствующем ответном сообщении имеет следующий формат:

(1) <wsman:RequestedEPR ...>

(2) [<wsa:EndpointReference>

(3)wsa:EndpointReferenceType

(4) </wsa:EndpointReference> |

(5) <wsman:EPRInvalid/> |

(6) <wsman:EPRUnknown/>]

(7) </wsman:RequestedEPR>

Следующие определения описывают дополнительные нормативные ограничения на приведенный выше формат сообщения:

wsman:RequestedEPR/wsa:EndpointReference

один из трех элементов, которые могут быть возвращены как дочерний элемент элемента wsman:RequestedEPR

Использование этого элемента указывает на то, что служба поняла просьбу возвратить EPR ресурса и включает EPR ресурса. Возвращенный EPR вычисляется после того, как произошли все намеренные эффекты или побочные эффекты связанного сообщения-запроса. EPR может не измениться в результате операции, но служба все же обязана возвратить его.

wsman:RequestedEPR/wsman:EPRInvalid один из трех элементов, которые могут быть возвращены как дочерний элемент элемента wsman:RequestedEPR

Использование этого элемента (никакое значение не требуется) указывает на то, что служба понимает просьбу возвратить EPR ресурса, но неспособна вычислить полный EPR. Однако служба в состоянии решить, что данный обмен сообщениями изменил представление ресурса таким способом, при котором любые предыдущие ссылки на ресурс более не действительны. Когда будет возвращен EPRInvalid, клиент не должен использовать в последующих операциях старый wsa:EndpointReference.

wsman:RequestedEPR/wsman:EPRUnknown

один из трех элементов, которые могут быть возвращены как дочерний элемент элемента wsman:RequestedEPR

Использование данного элемента (никакое значение не требуется) указывает на то, что служба понимает запрос возвратить EPR ресурса, но неспособна определить, действительны ли еще существующие ссылки на ресурс. Когда возвращен EPRUnknown, клиент может попытаться использовать в последующих операциях старый wsa:EndpointReference. Однако результат использования старого wsa:EndpointReference непредсказуем; результат может быть как отказом, так и успешным ответом.

7 Доступ к ресурсу

7.1 Общие положения

Доступ к ресурсу относится ко всем синхронным операциям, связанным с получением, изменением и перечислением значений. Подпункты раздела 7 определяют механизм для получения специфич-

ных для управления основанных на XML представлений сущностей, таких как управляемые ресурсы, посредством инфраструктуры Веб-служб.

Конкретно определены две операции для отправления и получения представления управления данного ресурса и две операции — для создания и удаления ресурса управления и его соответствующего представления. Получение ресурсов с множеством экземпляров достигается с использованием сообщения Enumerate. Данная спецификация не определяет сообщения или методы для пакетных операций, таких, как сгруппированные Get или Delete. Все такие операции можно отправить как серию однократных сообщений.

Следует отметить, что поддержание состояния ресурса по большей степени подчиняется политике «без гарантий» хостинг-сервера. Когда клиент получает от службы подтверждение запроса на создание или обновление ресурса, он может обоснованно ожидать, что ресурс теперь существует в подтвержденном местоположении и с подтвержденным представлением, но это не служит гарантией даже в отсутствие третьих сторон. Сервер может изменить представление ресурса, удалить ресурс полностью или может возвратить удаленный ресурс.

Например, сервер может хранить информацию о состоянии ресурса на внешнем диске. Если этот диск выходит из строя и сервер восстанавливает информацию о состоянии с ленты резервного копирования, то изменения, произошедшие после резервного копирования, будут потеряны.

У сервера могут быть другие эксплуатационные процессы, которые изменяют информацию о состоянии ресурса. Сервер может управлять фоновым процессом, который проверяет ресурсы на предмет нежелательного содержания и удаляет любые такие найденные ресурсы. Сервер может произвести чистку ресурсов, к которым не был получен доступ в течение определенного периода времени. Сервер может применить квоты для хранения, которые иногда заставляют его производить чистку ресурсов.

В сущности, подтверждение службой того, что запрос на создание, изменение или удаление ресурса был обработан, подразумевает обязательство только в момент, когда подтверждение было произведено. Несмотря на то, что, как правило, ресурсы должны быть долговечны и стабильны, нет никаких гарантий, и клиентам следует программировать осмотрительно.

Нет требований к однородности в представлениях ресурса между сообщениями, определенными в настоящем стандарте. Например, представления, требуемые от Create или Put, могут отличаться от представления, возвращенного Get, в зависимости от семантических требований службы. Кроме того, нет определенных требований к тому, чтобы содержимое ресурса было фиксированным для любой данной ссылки оконечной точки. Содержимое ресурса может меняться в зависимости от факторов окружения, таких как контекст защиты, время суток, конфигурация или динамическое состояние обслуживания.

Согласно модели обработки SOAP другие спецификации могут определить заголовки SOAP, которые можно опционально добавить в сообщение-запрос, чтобы потребовать передачу подмножества или применение преобразований ресурсов, связанного со ссылкой оконечной точки. При использовании URI действий, определенных данной спецификацией, такие дополнительные спецификации должны также обеспечивать основные модели обработки, определенные в данном документе.

Примечание — WSDL для операций доступа ресурса (см. Приложение G), а также псевдосхема и примеры фрагментов сообщений, приведенные в разделе 7, не применимы, как представлено, без предварительной замены текста «*resource-specific-GED*» специфичным для приложения GED.

Пример:

Ниже приведен полный пример гипотетического запроса Get:

(1) <s:Envelope

(2) xmlns:s="gost.e-ecolog.ru

(3) xmlns:wsa="gost.e-ecolog.ru

(4) xmlns:wsman="gost.e-ecolog.ru >

(5) <s:Header>

(6) <wsa:To>gost.e-ecolog.ru >

(7) <wsman:ResourceURI>gost.e-ecolog.ru

</wsman:ResourceURI>

(8) <wsa:ReplyTo>

(9) <wsa:Address>

(10) gost.e-ecolog.ru
(11) </wsa:Address>
(12) </wsa:ReplyTo>
(13) <wsa:Action>
(14) gost.e-ecolog.ru
(15) </wsa:Action>
(16) <wsa:MessageID>
(17) urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a87
(18) </wsa:MessageID>
(19) <wsman:SelectorSet>
(20) <wsman:Selector Name="LUN"> 2 </wsman:Selector>
(21) </wsman:SelectorSet>
(22) <wsman:OperationTimeout> PT30S </wsman:OperationTimeout>
(23) </s:Header>
(24) <s:Body/>
(25) </s:Envelope>

Обращаем внимание, что wsa:ReplyTo указывает, что ответ нужно отправить по тому же соединению, что и запрос (строка 10), действие Get (строка 14), и ResourceURI (строка 7) и wsman:SelectorSet (строка 20), они используются для того, чтобы обращаться к требуемой информации управления. Данный пример предполагает использование модели адресации WS-Management по умолчанию. Служба, как ожидается, закончит операцию через 30 секунд, либо вернет клиенту отказ (строка 22).

Кроме того, s:Body в запросе Get не имеет содержимого.

Пример:

Ниже показан гипотетический ответ на предыдущий гипотетический запрос Get:

(26) <s:Envelope
*(27) xmlns:s="*gost.e-ecolog.ru
*(28) xmlns:wsa="*gost.e-ecolog.ru
*(29) xmlns:wsman="*gost.e-ecolog.ru >
(30) <s:Header>
(31) <wsa:To>
(32) gost.e-ecolog.ru
(33) </wsa:To>
(34) <wsa:Action s:mustUnderstand="true">
(35) gost.e-ecolog.ru
(36) </wsa:Action>
(37) <wsa:MessageID s:mustUnderstand="true">
(38) urn:uuid:217a431c-b071-3301-9bb8-5f538bec89b8
(39) </wsa:MessageID>
(40) <wsa:RelatesTo>
(41) urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a87
(42) </wsa:RelatesTo>
(43) </s:Header>
(44) <s:Body>
(45) <PhysicalDisk
*xmlns="*gost.e-ecolog.ru >
(46) <Manufacturer> Acme, Inc. </Manufacturer>
(47) <Model> 123-SCSI 42 GB Drive </Model>
(48) <LUN> 2 </LUN>
(49) <Cylinders> 16384 </Cylinders>
(50) <Heads> 80 </Heads>

(51) <Sectors> 63 </Sectors>

(52) <OctetsPerSector> 512 </OctetsPerSector>

(53) <BootPartition> 0 </BootPartition>

(54) </PhysicalDisk>

(55) </s:Body>

(56) </s:Envelope>

Обратите внимание, что в ответе используется адрес wsa:To (строка 32), который исходный запрос определил ранее в wsa:ReplyTo. Кроме того, wsa:MessageID является уникальным для этого ответа (строка 38). wsa:RelatesTo (строка 41) содержит UUID wsa:MessageID исходного запроса для того, чтобы позволить клиенту соотнести запрос и ответ.

s:Body (строки 44—55) содержит требуемое представление ресурса.

Тот же самый общий подход существует для Delete, за исключением того, что в s:Body ответа отсутствует содержимое. Операции Create и Put аналогичны, за исключением того, что в s:Body запроса присутствует содержимое для определения созданного или обновленного значения.

7.2 Унификация адресации

Там, где это применимо, EPR ресурса может быть одинаковым независимо от того, используется ли операция Get, Delete или Put. Это не строгое требование, но оно уменьшает время обучения и подготовки, требуемое для построения и использования инструментов, поддерживающих WS-Managment.

Create — это особый случай, когда EPR недавно созданного ресурса зачастую неизвестен, пока ресурс фактически не создан. Например, хотя теоретически возможно возвратить информацию о выполняющемся процессе, используя в заголовке адресации гипотетическое ProcessID, как правило, в фазе создания невозможно назначить ProcessID, потому что нижележащая система не поддерживает это понятие. Таким образом операция Create будет иметь иные заголовки адресации, нежели соответствующие операции Get или Delete.

Если используется модель адресации WS-Management по умолчанию, обычно в качестве «типа» используется ResourceURI и значения селектора для идентификации «экземпляра». Таким образом, один и тот же адрес используется для Get, Put и Delete при работе с одним и тем же самым экземпляром. При перечислении всех экземпляров селекторы могут быть опущены и будет использоваться только ResourceURI для указания на «тип» перечисляемого объекта. Операция Create может также использовать этот сценарий или иметь свой собственный способ использовать ResourceURI и (или даже не использовать) селекторы. Данный шаблон не является требованием.

Везде в тексте предполагается, что сообщения s:Body содержат XML с правильными и действительными пространствами имен XML, указывающими на XML-схемы, которые могут использоваться для валидации сообщения. Большинство служб и клиентов не выполняют валидацию сообщений в реальном времени в эксплуатационном режиме из-за ограничений на производительность; однако во время отладки или другой верификации систем можно разрешить валидацию, и сообщения без соответствующих объявлений пространства имен XML будут считаться недействительными.

При выполнении операции доступа к ресурсу могут произойти побочные эффекты. Например, удаление конкретного ресурса с помощью Delete может привести к нескольким другим зависимым исчезновениям экземпляров, а операция Create может привести к логическому созданию более чем одного ресурса, которые могут быть впоследствии возвращены посредством операции Get. Таким же образом операция Put может привести к переименованию целевого экземпляра, переименованию некоторого несвязанного экземпляра или удалению некоторого несвязанного экземпляра. Эти побочные эффекты определяются службой, и данная спецификация не делает заявлений о классификации и семантике объектов, ккоторым применяются такие операции.

7.3 Операция Веб-службы Get

Операция Веб-службы (Get) определена для получения однократного снимка представления ресурса. Снимок — это полное представление XML ресурса во время обработки запроса службой.

Сообщение-запрос Get должно иметь следующую форму:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:MessageID>xs:anyURI</wsa:MessageID>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body .../>

(11) </s:Envelope>

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный требуемый элемент должен содержать значение

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

Запрос Get должен быть направлен ресурсу, чье представление требуется получить.

По умолчанию для запроса Get не определены блоки тела. Согласно модели обработки SOAP другие спецификации могут ввести различные типы расширений по семантике этого сообщения, которые разрешены посредством заголовков, отмеченных s:mustUnderstand = «true». Такие расширения могут определить, как ресурс или подмножества должны быть получены или преобразованы перед получением. Спецификации, которые определяют такие расширения, должны позволить обработку базового запроса Get без этих расширений. Поскольку может оказаться невозможным отправить ответ оригинальному отправителю, дополнительным спецификациям следует рассмотреть добавление в ответ соответствующего значения заголовка SOAP как сигнал приемнику, что используется расширение.

Реализации могут ответить сообщением об отказе, используя стандартные коды отказа, определенные в адресации (например, wsa:ActionNotSupported). Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если ресурс принимает запрос Get, он должен отправить ответ следующей формы:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body ...>

(11) содержимое, зависящее от ресурса

(12) </s:Body>

(13) </s:Envelope>

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение gost.e-ecolog.ru
transfer/GetResponse. Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/child

Само представление должно быть дочерним элементом элемента SOAP:Body в ответном сообщении.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Операция Get получает представления ресурса. Сообщение может быть направлено для того, чтобы вернуть сложный XML-документ или вернуть единственное простое значение. Характер и сложность представления не ограничены данной спецификацией.

R7.3-1 Совместимой службе следует поддерживать операции Get для обслуживания запросов метаданных о самой службе или проверки результата предыдущего действия или операции.

Это утверждениене ограничивает реализации от предоставления дополнительных сходных методов для поиска метаданных и ресурса.

R7.3-2 Выполнение Get не следует само по себе оказывать побочного воздействия на значение ресурса.

R7.3-3 Если объект не может быть получен из-за блокировок, одновременного доступа или иных подобных конфликтов, следует вернуть отказ wsman:Concurrency.

На практике Get разработан, чтобы вернуть XML, который соответствует реальным объектам. Чтобы получить значения отдельных свойств, либо клиент может обработать содержимое XML для получения требуемого значения, либо служба может поддерживать доступ на уровне фрагмента (см. 7.7).

Использование отказов в целом соответствует описанному в разделе 14. Невозможность определить местонахождение или получить доступ к ресурсу эквивалентна проблемам с сообщением SOAP, когда выявлена некорректная EPR. Нет никаких особых отказов для Get.

7.4 Операция Веб-сервиса Put

Операция Веб-сервиса (Put) определена для обновления ресурса посредством предоставления представления замены. Ресурс может принимать обновления с представлением XML, отличающимся от возвращаемых ресурсов; в таком случае семантика операции по обновлению определяется ресурсом.

Сообщение-запрос Put должно иметь следующую форму:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:MessageID>xs:anyURI</wsa:MessageID>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body...>

(11) элемент-специфичный-для-ресурса

(12) </s:Body>

(13) </s:Envelope>

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения: /s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение % /s:Envelope/s:Body/child

Представление, которое будет использоваться для обновления, должно быть дочерним элементом для элемента s:Body сообщения-запроса.

Запрос Put должен быть направлен ресурсу, представление которого требуется заменить. Согласно модели обработки SOAP, другие спецификации могут ввести различные типы расширений к данному сообщению, которые представлены посредством заголовков, помеченных s:mustUnderstand= «true». Такие расширения могут потребовать, чтобы полное или частичное обновление осуществлялось с использованием символических, основанных на инструкциях или других методологиях.

Спецификации расширений могут также определить расширения к оригинальному запросу Put, обеспеченные дополнительными заголовками SOAP, которые управляют характером ответа (см. информацию о PutResponse ниже в данном разделе).

Спецификации, которые определяют любое из этих расширений, должны позволять обработку сообщения Put без таких расширений.

В дополнение к стандартным кодам отказа, определенным в адресации, в реализациях может использоваться код отказа wsmt:InvalidRepresentation, если представленное представление недействительно для целевого ресурса. Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Успешная операция Put обновляет текущее представление, связанное с целевым ресурсом.

Если ресурс примет запрос Put и выполнит требуемое обновление, то он должен отправить ответ следующей формы:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body ...>

(11) элемент-специфичный-для-ресурса ?

(12) </s:Body>

(13) </s:Envelope>

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение gost.e-ecolog.ru. Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/child

Реализация службы должна заранее выбрать, возвратить ли пустое Body или полученное представление ресурса. Данный выбор должен быть явным образом заявлен в WSDL, если WSDL будет предоставлен.

По умолчанию служба должна возвратить текущее представление ресурса как дочерний элемент s:Body, если обновленное представление будет отличаться от представления, отправленного в сообщении запроса Put.

Как оптимизация и как услуга запросчику, элемент s:Body ответного сообщения следует возвращать пустым, если обновленное представление не отличается от представления, отправленного в сообщении Put запроса; то есть, если служба приняла новое представление дословно.

Такой ответ (пустой s:Body) подразумевает, что запрос обновления прошел успешно полностью (предполагая, что в процессе выполнения не происходили никакие преобразования). Однако служба может возвратить текущее представление ресурса как начальный дочерний элемент s:Body даже в этом случае.

Дополнительные спецификации могут определить расширения к первоначальному запросу Put, которому присвоены дополнительные значения заголовка для оптимизации ответа. В отсутствие таких заголовков поведение должно быть таким, как описано ранее. Спецификации, которые определяют любое из этих расширений, должны разрешать обработку сообщений без таких расширений. Поскольку может оказаться невозможным отправить ответ оригинальному отправителю, дополнительным спецификациям следует рассмотреть добавление в ответ соответствующего значения заголовка SOAP как сигнал приемнику, что используется расширение.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если ресурс можно обновить полностью в рамках ограничений соответствующей XML-схемы для ресурса, служба может поддерживать операцию Put.

R7.4-1 Служба, соответствующая настоящему стандарту, может поддерживать Put.

R7.4-2 Если единственный экземпляр ресурса может быть обновлен (в рамках ограничений своей схемы) при помощи сообщения SOAP и этот ресурс впоследствии может быть получен с использованием Get, служба должна поддерживать обновление ресурса при помощи Put. Служба может дополнительно предоставлять для обновлений специальный метод.

R7.4-3 Если единственный экземпляр ресурса содержит совокупность модифицируемых и немодифицируемых свойств, сообщение Put может содержать значения как для модифицируемых, так и для немодифицируемых свойств, если содержимое XML является правомерным относительно своего пространства имен XML-схемы. Если сообщение Put будет содержать значения для модифицируемых свойств, то служба должна установить данные свойства в эти значения во время операции Put. Если сообщение Put содержит значения для немодифицируемых свойств, службе следует проигнорировать эти значения во время операции Put. Если ни одно из свойств не является модифицируемым, следует вернуть сообщение-отказ wsa:ActionNotSupported.

Эта ситуация, как правило, происходит, если выполнена операция Get, полученное значение изменено и все обновленное представление отправлено с использованием Put. В этом случае могут присутствовать любые значения только для чтения.

Поскольку Put содержит для экземпляра полное новое представление, возникают трудности. Если схема ресурса требует присутствия какого-либо определенного значения (minOccurs не равно нулю), оно будет доставлено как часть сообщения Put, даже если исходное значение не будет изменено.

R7.4-4 Если операция Put определяет модифицированное значение как NULL, используя атрибут xsi:nil, то служба должна установить значение в NULL.

Если определение схемы включает элементы, которые являются необязательными (minOccurs=0), сообщение Put может опустить эти значения. Существующие реализации обеспечивают два различных ответа в случае, когда эти элементы являются модифицируемыми. Они устанавливают значение опущенного элемента NULL или оставляют значение неизменным. Учитывая эту действительность, применяются следующие правила:

R7.4-5 Любые модифицируемые свойства, которые являются необязательными в XML-схеме (то есть minOccurs ="0") и которые опущены в сообщении Put, или будут установлены в определенное для ресурса значение по умолчанию, или оставлены неизменными. Рекомендуется установка в определенное ресурсом значение по умолчанию.

Примечания

1 Неустановленные элементы могут иметь собственные значения в результате других ограничений.

2 Определенное ресурсом значение по умолчанию выходит за рамки настоящего стандарта.

Для того чтобы избежать предоставления всех значений при обновлении изолированных значений, используйте механизм доступа к ресурсу на уровне фрагмента, описанный в 7.7.

Вкратце, элемент s:Body сообщения Put удовлетворяет ограничениям связанной XML-схемы.

Примеры:

1 Предположим, что Get возвращает следующую информацию:

(1) <s:Body>

(2) <MyObject xmlns="examples.org/2005/02/MySchema">

(3) <A> 100 </A>

(4) <B> 200 </B>

(5) <C> 100 </C>

(6) </MyObject>

(7) </s:Body>

2 Соответствующая XML-схема определила A, B, и C как minOccurs=1:

(8) <xs:element name="MyObjecct">

(9) <xs:complexType>

(10) <xs:sequence>

(11) <xs:element name="A" type="xs:int" minOccurs="1" maxOccurs="1"/>

(12) <xs:element name="B" type="xs:int" minOccurs="1" maxOccurs="1"/>

(13) <xs:element name="C" type="xs:int" minOccurs="1" maxOccurs="1"/>

(14) ...

(15) </xs:sequence>

(16) </xs:complexType>

(17) </xs :element>

В этом случае соответствующему Put нужно содержать все три элемента, потому что схема декларирует, что все три присутствовуют. Даже если клиенту нужно обновить только значение <B>, клиент должен предоставить все три значения. Это обычно означает, что клиент сначала должен отправить Get, чтобы сохранить текущие значения <A> и <C>, изменить <B> на требуемое значение и затем переписать объект с помощью Put. Как отмечено в R7.4-3, служба может проигнорировать попытки обновить значения, которые предназначены только для чтения относительно нижележащего реального объекта.

R7.4-6 Совместимой службе следует поддерживать Put, используя ту же самую EPR, что и в операции Get или других сообщениях, если механизм Put для ресурса семантически не отличим.

R7.4-7 Если в поставляемом элементе Body нет корректного содержимого для обновления ресурса, следует вернуть сообщение-отказ wsmt:InvalidRepresentation со следующим кодом детализации:

- если какие-либо значения в s:Body неверны:

gost.e-ecolog.ru

- если какие-либо значения в s:Body отсутствуют:

gost.e-ecolog.ru

- если используется неверное пространство имен XML-схемы и не распознано службой:

gost.e-ecolog.ru

R7.4-8 Если объект не может быть обновлен из-за условий блокировки, одновременного доступа или подобных конфликтов, следует вернуть сообщение-отказ wsman:Concurrency.

R7.4-9 Операция Put может привести к изменению EPR для ресурса, так как обновляемые значения могут, в свою очередь, вызвать изменение идентичности.

Поскольку службы WS-Management, как правило, делегируют Put в нижележащие подсистемы, служба не всегда может знать об изменении идентичности. Клиенты могут использовать механизм из 6.5, чтобы получить информацию об изменениях EPR, которые, возможно, произошли как побочный эффект выполнения операции Put.

R7.4-10 Службе рекомендуется возвращать новое представление в ответе на Put во всех случаях. Зачастую бывает трудно узнать, отличается ли фактическое получившееся представление от требуемого обновления, так как у реализаций может быть недостаточно ресурсов для определения эквивалентности требуемого обновления с фактическим получившимся представлением.

Смысл этого правила в том, что, если новое представление не возвращено, оно точно соответствует тому, что было представлено в сообщении Put. Поскольку реализации могут редко гарантировать это, они могут всегда возвращать новое представление.

R7.4-11 Если нельзя отправить сообщение об успехе операции из-за ограничений кодирования или по другим причинам, как описано в данном пункте, и она не может быть полностью отменена до состояния до начала операции, следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

R7.4-12 Операция Put может содержать обновления нескольких значений. Служба должна успешно выполнить обновление всех указанных значений или вернуть отказ, который был причиной ошибки. Если какой-либо отказ возвращен, подразумевается, что из *n* возможных значений обновлены 0...*n*-1.

7.5 Delete

Данная спецификация определяет одну операцию (Delete) Веб-службы для полного удаления ресурса.

Дополнительные спецификации могут определить расширения к запросу Delete, с дополнительными значениями заголовка, которые специфически управляют предварительными условиями для успешного завершения Delete и которые могут управлять характером или форматом ответа. Поскольку может оказаться невозможным отправить ответ оригинальному отправителю, дополнительным спецификациям следует рассмотреть добавление в ответ соответствующего значения заголовка SOAP как сигнал приемнику, что используется расширение.

Сообщение-запрос Delete должно иметь следующую форму:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:MessageID>xs:anyURI</wsa:MessageID>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body ... />

(11) </s:Envelope>

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение gost.e-ecolog.ru
transfer/Delete. Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

Запрос Delete должен быть направлен ресурсу, который будет удален.

Для запроса Delete не определены блоки тела.

Реализации могут ответить сообщением отказа, используя стандартные коды отказа, определенные в адресации (например, wsa:ActionNotSupported). Другие компоненты приведенной выше схемы не ограничены данной спецификацией.

Успешная операция Delete делает неправомерным текущее представление, связанное с предназначенным ресурсом.

Если ресурс примет запрос Delete, то он должен дать ответ по следующей форме:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body .../>

(11) </s:Envelope>

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение gost.e-ecolog.ru
transfer/DeleteResponse. Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

По умолчанию для ответа Delete не определены блоки s:Body. Спецификации, которые определяют расширения для использования в оригинальном запросе Delete, управляющие форматом ответа, должны позволить обработку сообщения Delete без таких расширений.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

В целом, для единообразия адресация может совпасть с адресацией для операции Get, но это не является абсолютным требованием.

R7.5-1 Служба, соответствующая настоящему стандарту, может поддерживать Delete.

R7.5-2 Совместимой службе следует поддерживаать Delete с использованием того же самого EPR, что и соответствующее сообщение Get и другие сообщения, если механизм удаления для ресурса не является семантически иным.

R7.5-3 Если удаление поддерживается и соответствующий ресурс может быть получен посредством Get, совместимой службе следует поддерживать удаление посредством Delete. Служба может дополнительно предоставлять для удаления специализированное действие.

R7.5-4 Если объект нельзя удалить из-за условий блокировки, одновременного доступа или иных подобных конфликтов, следует вернуть отказ wsman:Concurrency.

На практике операция Delete удаляет экземпляр ресурса из области видимости клиента и является логическим удалением.

Такая операция может привести к фактическому удалению, такому как удаление кортежа из таблицы базы данных, или может имитировать удаление, отделив представление от реального объекта. Например, удаление «принтера» не приводит к буквальному уничтожению принтера, но просто удаляет его из области доступа обслуживания или «отключает» его от таблиц именования. WS-Management не делает различия между фактическими удалениями и логическими удалениями.

Чтобы удалить отдельные значения свойств в пределах объекта, который сам по себе не должен быть удален, клиент может выполнить Put согласно подразделу 7.4, либо служба может поддерживать Delete на уровне фрагмента (см. 7.7).

Использование отказов в целом соответствует описанному в главе 14. Невозможность определить местонахождение или получить доступ к ресурсу эквивалентна проблемам с сообщением SOAP, когда выявлена некорректная EPR. Нет никаких особых отказов для Delete.

7.6 Создание ресурса (Create)

Операция Веб-службы (Create) определена для создания ресурса и обеспечения его начального представления. В некоторых случаях начальное представление может быть представлением логического конструктора для ресурса и может, таким образом, отличаться структурно от представления, возвращенного Get или требуемого Put. Это различие происходит из-за того, что требование параметризации для создания ресурса зачастую отличается от установившегося представления ресурса. Реализации должны обеспечить метаданные, которые описывают использование представления и как оно касается созданного ресурса, но такие механизмы выходят за рамки настоящего стандарта. Фабрика ресурса, которая получает запрос Create, выделяет новый ресурс, который инициализируется от представленного представления. Новому ресурсу назначают определенную службой ссылку оконечной точки, которая возвращается в ответном сообщении.

Сообщение-запрос Create должно иметь следующую форму:

(1) ***<s:Envelope ...>***

(2) ***<s:Header ...>***

(3) ***<wsa:Action>***

(4) gost.e-ecolog.ru

(5) ***</wsa:Action>***

(6) ***<wsa:MessageID>xs:anyURI</wsa:MessageID>***

(7) ***<wsa:To>xs:anyURI</wsa:To>***

(8) ***...***

(9) ***</s:Header>***

(10) ***<s:Body ...>***

(11) ***элемент-специфичный-для-ресурса***

(12) ***</s:Body>***

(13) ***</s:Envelope>***

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение gost.e-ecolog.ru. Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/child

Дочерний элемент s:Body не должен быть опущен. Содержание этого элемента зависит от службы и может содержать буквальное начальное представление ресурса, представление конструктора для ресурса или другие инструкции для создания ресурса.

Дополнительные спецификации могут также определить расширения базового запроса Create, разрешаемые дополнительными заголовками SOAP, которые ограничивают характер ответа (см. информацию о CreateResponse ниже в этом пункте). Аналогично они могут требовать заголовки, которые управляют интерпретацией s:Body как части процесса создания ресурса.

Такие спецификации должны также позволять обрабатывать сообщение Create без таких расширений.

Запрос Create должен быть направлен фабрике ресурса, способной создавать желаемый новый ресурс. Эта фабрика отличается от создаваемого ресурса (который по определению не существует до успешной обработки сообщения-запроса Create).

В дополнение к стандартным кодам отказа, определенным в адресации, реализации могут использовать код отказа wsmt:InvalidRepresentation, если представленное представление для целевого ресурса недействительно.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если фабрика ресурса принимает запрос Create, то она должна дать ответ следующей формы:

(1)<s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body ...>

(11) <wsmt:ResourceCreated>ссылка-на-оконечную-точку</wsmt:ResourceCreated>

(12) </s:Body>

(13) </s:Envelope>

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение gost.e-ecolog.ru. Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/wsmt:ResourceCreated

Данный обязательный элемент должен содержать ссылку ресурса для новосозданного ресурса. Эта ссылка ресурса, представленная как ссылка оконечной точки, как определено в адресации, должна идентифицировать ресурс для будущих операций Get, Put и Delete.

Дополнительные спецификации могут определить расширения к первоначальному запросу Create, разрешенные дополнительными значениями заголовка. Эти заголовки могут переопределять поведение по умолчанию, если они отмечены s:mustUnderstand = «true». В отсутствие таких дополнительных заголовков поведение должно быть таким, как описано в предыдущих пунктах. Поскольку может оказаться невозможным отправить ответ оригинальному отправителю, дополнительным спецификациям следует рассмотреть добавление в ответ соответствующего значения заголовка SOAP как сигнал приемнику, что используется расширение.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

В целом, адресация отличается от той, что использовалась для Get или Delete, в том, что EPR, назначенный недавно созданному экземпляру для последующего доступа, не обязательно является частью содержания XML, используемого для создания ресурса. Поскольку EPR обычно назначается службами или одной из их нижележащих систем, CreateResponse содержит применимый EPR для недавно созданного экземпляра.

R7.6-1 Служба, соответствующая настоящему стандарту, может поддерживать Create.

R7.6-2 Если может быть создан единичный ресурс с использованием сообщения SOAP, и этот ресурс может быть впоследствии полученс использованием Get, то службе следует поддерживать создание ресурса с использованием Create. Служба может дополнительно предоставлять специализированный метод создания экземпляров.

R7.6-3 Если поставляемое SOAP Body не имеет правильного содержимого для ресурса, который должен быть создан, то следует вернуть сообщение-отказ wsmt:InvalidRepresentation и коды детализации следующим образом:

- если одно или более значений в <s:Body> были неправильны:

gost.e-ecolog.ru

- если одно или более значений в <s:Body> отсутствовали:

gost.e-ecolog.ru

- если использовалось неправильное пространство имен XML-схемы и оно не признано службой:

gost.e-ecolog.ru

R7.6-4: Служба не должна использовать Create, чтобы изменить значение существующего представления (за исключением указанного в 7.11). Если предназначенный объект уже существует, следует вернуть сообщение-отказ wsman:AlreadyExists.

Не требуется, чтобы в теле сообщения Create использовалась та же схема, что и для возвращаемого значения операции Get для ресурса. Часто значения, требуемые для создания ресурса, отличаются от тех, которые возвращаются операцией Get или используются для обновлений операцией Put.

Если службе необходимо поддерживать создание отдельных значений в пределах представления (создание на уровне фрагмента, вставка в массив и т. д.), она может поддерживать доступ на уровне фрагмента (см. 7.7).

R7.6-5 Ответ на сообщение Create должен содержать новый EPR созданного ресурса в элементе ResourceCreated.

R7.6-6 Это правило преднамеренно оставлено незаполненным.

Пример:

Ниже приведен гипотетический пример ответа для недавно созданного виртуального диска:

(1) ***<s:Envelope***

(2) ***xmlns:s="***gost.e-ecolog.ru

(3) ***xmlns:wsa="***gost.e-ecolog.ru ***"***

(4) ***xmlns:wsman="***gost.e-ecolog.ru

(5) ***xmlns:wsmt="***gost.e-ecolog.ru ***>***

(6) ***<s:Header>***

(7) ***...***

(8) ***<wsa:Action>***

(9) gost.e-ecolog.ru

(10) ***</wsa:Action>***

(11) ***...***

(12) ***</s:Header>***

(13) ***<s:Body>***

(14) ***<wsmt:ResourceCreated>***

(15) ***<wsa:Address>***

(16) gost.e-ecolog.ru

(17) ***</wsa:Address>***

(18) ***<wsa:ReferenceParameters>***

(19) ***<wsman:ResourceURI>***

(20) gost.e-ecolog.ru

(21) ***</wsman:ResourceURI>***

(22) ***<wsman:SelectorSet>***

(23) ***<wsman:Selector Name="ID"> F: </wsman:Selector>***

(24) ***</wsman:SelectorSet>***

(25) ***</wsa:ReferenceParameters>***

(26) ***</wsmt:ResourceCreated>***

(27) ***</s:Body>***

(28) ***</s:Envelope>***

Данный пример предполагает, что используется модель адресации по умолчанию. Ответ содержит блок ResourceCreated (строки14—26), который содержит новую ссылку оконечной точки созданного ресурса, включая его ResourceURI и SelectorSet. Данный адрес будет использован, чтобы получить ресурс в последующей операции Get.

Служба может использовать сетевой адрес, который совпадает с адресом <wsa:To> в запросе Create.

R7.6-7 Служба может проигнорировать любые значения в начальном представлении, которые служат только для чтения с точки зрения нижележащего реального объекта.

Это правило позволяет Get, Put и Create использовать одну и ту же схему. Put также позволяет службе игнорировать свойства «только для чтения» во время обновления.

R7.6-8 Если нельзя сообщить об успехе операции, как описано в этом пункте, а также если его нельзя отменить, то следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

7.7 Доступ на уровне фрагмента

Поскольку механизм доступа к ресурсу, определенный в настоящем стандарте, работает с экземплярами целиком, и может быть неудобно определять сотни или тысячи EPR только для того, чтобы моделировать доступ на уровне фрагмента полными ERP, WS-Management поддерживает понятие доступа на уровне фрагмента (свойств) для ресурсов, доступ к которым обычно получают посредством операций доступа к ресурсу. Данный доступ осуществляется посредством специального использования этих операций.

Из-за ограничений XML-схемы, которые приведены в 7.6, часто бывает неверным просто возвращать подмножество XML, определенного для объекта, к которому производится доступ, так как подмножество может нарушить XML-схему для этого фрагмента. Для поддержки доступа к фрагментам или отдельным элементам объекта представления внесены некоторые изменения в основные операции доступа к ресурсу.

R7.7-1 Служба, соответствующая настоящему стандарту, может поддерживать доступ на уровне фрагмента. Если служба поддерживает доступ на уровне фрагмента, то служба не должна вести себя, как если бы выполнялись нормальные операции доступа, но должна воздействовать исключительно на определенные фрагменты. Если служба не поддерживает доступ на уровне фрагмента, то она должна вернуть сообщение-отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

R7.7-2 Совместимая служба, которая поддерживает доступ на уровне фрагмента, должна принимать следующий заголовок SOAP во всех запросах и включить его во все ответы, которые транспортируют фрагменты:

(1) <wsman:FragmentTransfer s:mustUnderstand = “true”>

(2) xpath к фрагменту

(3) </wsman:FragmentTransfer>

Значением этого заголовка является выражение XPath 1.0, которое идентифицирует передаваемый фрагмент по отношению к полному представлению объекта. Если используется выражение, отличное от XPath 1.0, можно добавить атрибут Dialect для указания на этот факт:

(1) <wsman:FragmentTransfer s:**mustUnderstand = “true”**

(2) Dialect=“URIToNewFragmentDialect”>

(3) выражение диалекта

(4) </wsman:FragmentTransfer>

Клиент должен понимать, что если заголовок не отмечен mustUnderstand = «true», служба может обработать запрос, игнорируя заголовок, что может привести к неожиданным и потенциально серьезным побочным эффектам.

XPath явно определен как диалект из-за его важности, однако в реализации не требуется поддерживать XPath как диалект фрагмента. Разрешен любой другой тип языка для описания доступа на уровне фрагмента при условии, что установлено значение Dialect для указания службе, какой диалект используется.

R7.7-3 Для операций к ресурсу на уровне фрагмента, которые используют [XPath 1.0] (URI диалекта gost.e-ecolog.ru), значением элемента /s:Envelope/s:Header/wsman:FragmentTransfer будет выражение XPath. Это выражение Xpath вычисляется с использованием следующего контекста:

- узел контекста: корневой элемент представления XML ресурса, адресованного в запросе, который возвратила бы операция Get как начальный дочерний элемент элемента SOAP Body ответа, если операция Get была бы применена к адресованному ресурсу без использования доступа фрагмента;
- положение в контексте: 1;
- размер контекста: 1;
- привязки переменных контекста: нет;
- библиотеки функций: Основная библиотека функции [XPath 1.0];
- объявления пространств имен: свойство [in-scope namespaces] [Инфо-набор XML] элемента /s:Envelope/s:Header/wsman:FragmentTransfer запроса.

Это правило означает, что XPath должен интерпретироваться относительно представления XML ресурса, а не относительно произвольного содержимого SOAP.

Для операций EnumarationXPath интерпретируется как определено в разделе 8, несмотря на то, что результат впоследствии оборачивается в обертки

wsman:XmlFragment после оценки XPath.

Значение XPath может относиться ко всему узлу, таким образом, понятие фрагмента включает весь объект, тем самым операции доступа на уровне фрагмента составляют надмножество нормальных операций доступа к ресурсу.

Если полный синтаксис выражений XPath не может поддерживаться, с этой целью в приложении В описано общее подмножество. Однако в этих случаях по-прежнему в качестве URI диалекта указывается URI XPath.

R7.7-4 Если служба понимает доступ на уровне фрагмента, но не понимает указанный Dialect URI фрагмента либо диалект по умолчанию, служба должна вернуть отказ wsman:FragmentDialectNotSupported.

R7.7-5 Все сообщения доступа к ресурсу с фрагментами XML в любом направлении должны быть обернуты оберткой <wsman:XmlFragment>, которая содержит определение, подавляющее валидацию и позволяет проходить любому содержанию. Служба должна отклонить любую попытку использовать wsman:FragmentTransfer, если содержимое s:Body не обернуто элементом wsman:XmlFragment. В противном случае служба должна сбросить запрос с помощью отказа wsmt:InvalidRepresentation со следующим кодом детализации:

gost.e-ecolog.ru

Доступ к фрагменту может произойти на любом уровне, включая одинарный элемент, составные элементы, простые значения и атрибуты. На практике службы, как правило, поддерживают только доступ на уровне значения элементов.

R7.7-6 Если поддерживается доступ на уровне фрагмента, совместимой службе следует поддерживать, по крайней мере, доступ на уровне значения листового узла, используя выражение XPath, которое использует проверку узла */text()*. В этом случае значение не оборачивается в XML и передается непосредственно как текст в рамках обертки wsman:XmlFragment.

В сущности, передаваемое содержимое — это то, что возвращает операция XPath в полном XML.

R7.7-7 Если поддерживается доступ на уровне фрагмента, но выражение фильтра превышает возможности службы, следует вернуть сообщение-отказ wsman:CannotProcessFilter с текстом, объясняющим, почему фильтр вызвал проблему.

R7.7-8 Для всех операций на уровне фрагмента не разрешены частичные успехи. Значения выражения XPath или другого диалекта должны выполняться службами в полном объеме во всех операциях, а весь фрагмент, который был определен, должен успешно передаваться в любом направлении. Иначе, происходят отказы, как если бы ни одна из операции не преуспела.

Все отказы совпадают с отказами для нормальных, «полных» операций доступа к ресурсу.

Следующие пункты показывают, как изменяются основные операции доступа к ресурсу при передаче фрагментов XML.

7.8 Get на уровне фрагмента

Get на уровне фрагмента подобна полному Get за исключением заголовка wsman:FragmentTransfer.

Пример:

Следующий пример взят частично из примера 7.1:

```
(1)  <s:Envelope
(2)  xmlns:s="gost.e-ecolog.ru
(3)  xmlns:wsa="gost.e-ecolog.ru                                    "
(4)  xmlns:wsman="gost.e-ecolog.ru                                  >
(5)  <s:Header>
(6)  <wsa:To>
(7)  http://1.2.3.4/wsman
(8)  </wsa:To>
(9)  <wsman:ResourceURI>gost.e-ecolog.ru
     </wsman:ResourceURI>
(10) <wsa:ReplyTo>
(11) <wsa:Address>
(12) gost.e-ecolog.ru
(13) </wsa:Address>
(14) </wsa:ReplyTo>
(15) <wsa:Action>
(16) gost.e-ecolog.ru
(17) </wsa:Action>
(18) <wsa:MessageID>
(19) urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a87
(20) </wsa:MessageID>
(21) <wsman:SelectorSet>
(22) <wsman:Selector Name="LUN"> 2 </wsman:Selector>
(23) </wsman:SelectorSet>
(24) <wsman:OperationTimeout> PT30S </wsman:OperationTimeout>
(25) <wsman:FragmentTransfer s:mustUnderstand="true">
(26) Manufacturer
(27) </wsman:FragmentTransfer>
(28) </s:Header>
(29) <s:Body/>
(30) </s:Envelope>
```

В этом случае служба применяет указанное выражение XPath к представлению, которое было бы получено в обычном случае, и затем вместо этого возвращает фрагмент.

Пример:

Служба повторяет элемент wsman:FragmentTransfer в GetResponse (строки 48—50), чтобы указать фрагмент и показать, что фрагмент был передан. Ответ обернут в обертку wsman:XmlFragment, которая подавляет валидацию на соответствие схеме, которая бы применялась бы в ином случае.

```
(31)  <s:Envelope
(32)    xmlns:s="gost.e-ecolog.ru
(33)    xmlns:wsa="gost.e-ecolog.ru
(34)    xmlns:wsman="gost.e-ecolog.ru                                >
(35)   <s:Header>
(36)    <wsa:To>
(37)      gost.e-ecolog.ru
(38)    </wsa:To>
(39)    <wsa:Action s:mustUnderstand="true">
```

(40) gost.e-ecolog.ru
(41) **</wsa:Action>**
(42) **<wsa:MessageID s:mustUnderstand="true">**
(43) **urn:uuid:1a7e7314-d791-4b4b-3eda-c00f7e833a8c**
(44) **</wsa:MessageID>**
(45) **<wsa:RelatesTo>**
(46) **urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a87**
(47) **</wsa:RelatesTo>**
(48) **<wsman:FragmentTransfer s:mustUnderstand="true">**
(49) **Manufacturer**
(50) **</wsman:FragmentTransfer>**
(51) **</s:Header>**
(52) **<s:Body>**
(53) **<wsman:XmlFragment**
xmlns="gost.e-ecolog.ru **>**
(54) **<Manufacturer> Acme, Inc. </Manufacturer>**
(55) **</wsman:XmlFragment>**
(56) **</s:Body>**
(57) **</s:Envelope>**

Результат (строки 53—55) похож на тот, который возвращается типичным процессором XPath.

Чтобы получить собственно значение, без обертки элемента XML, клиент может использовать средства XPath, такие как оператор text().

Пример:

В следующем примере используется text(), чтобы получить имя изготовителя:

(1)<wsman:FragmentTransfer s:mustUnderstand = "true">
(2) Manufacturer/text()
(3)</wsman:FragmentTransfer>

Данный запрос приводит к следующему XML в SOAP Body ответа:

1)<wsman:XmlFragment>
(2) Acme, Inc.
(3)</wsman:XmlFragment>

7.9 Put на уровне фрагмента

Put на уровне фрагмента действует как обычный Put, за исключением того, что он передает только обновляемую часть. Хотя фрагмент можно считать частью экземпляра с точки зрения наблюдателя, фрагмент, на который ссылаются, во время выполнения операции рассматривают как «экземпляр».

Примечание — Put *всегда* является операцией по обновлению существующего элемента, будь это простым элементом или массивом. Чтобы создать или вставить новые элементы, требуется Create.

Пример:

Рассмотрим следующий XML в иллюстративных целях:

(1) **<a>**
(2) ****
(3) **<c></c>**
(4) **<d></d>**
(5) ****
(6) **<e>**
(7) **<f></f>**
(8) **<g></g>**
(9) **</e>**
(10) ****

Хотя <a> является полным представлением экземпляра ресурса, если операция ссылается на узел a/b во время операции Put, используя выражение XPath “b”, то содержимое <b> обновляется, не касаясь других частей <a>, таких как <e>. Если клиент хочет обновить только <d>, тогда используемое выражение XPath является “b/d”.

Примеры:

1 Продолжая пример в 7.1, если клиент хочет изменить значение <BootPartition>с 0 на 1, службе можно отправить следующий фрагмент Put:

```
(1)  <s:Envelope
(2)    xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(3)    xmlns:wsa="gost.e-ecolog.ru
(4)    xmlns:wsman="gost.e-ecolog.ru                                    >
(5)   <s:Header>
(6)    <wsa:To>
(7)     http://1.2.3.4/wsman
(8)    </wsa:To>
(9)    <wsman:ResourceURI>gost.e-ecolog.ru
       </wsman:ResourceURI>
(10)   <wsa:ReplyTo>
(11)    <wsa:Address>
(12)      gost.e-ecolog.ru
(13)    </wsa:Address>
(14)   </wsa:ReplyTo>
(15)   <wsa:Action>
(16)     gost.e-ecolog.ru
(17)   </wsa:Action>
(18)   <wsa:MessageID>
(19)    urn:uuid:d9726315-bc91-2222-9ed8-c044c9658a87
(20)   </wsa:MessageID>
(21)   <wsman:SelectorSet>
(22)    <wsman:Selector Name="LUN"> 2 </wsman:Selector>
(23)   </wsman:SelectorSet>
(24)   <wsman:OperationTimeout> PT30S </wsman:OperationTimeout>
(25)   <wsman:FragmentTransfer s:mustUnderstand="true">
(26)    BootPartition
(27)   </wsman:FragmentTransfer>
(28)  </s:Header>
(29)  <s:Body>
(30)  <wsman:XmlFragment>
(31)  <BootPartition> 1 </BootPartition>
(32)  </wsman:XmlFragment>
(33)  </s:Body>
(34) </s:Envelope>
```

2 Обертка <BootPartition> присутствует, так как она входит в значение XPath. Если бы в качестве выражения использовалось “BootPartition/text()”, то Body содержал бы простое значение, как в следующем примере:

```
(35)  <s:Header>
(36)   ...
(37)  <wsman:FragmentTransfer s:mustUnderstand="true">
(38)  BootPartition/text()
(39)  </wsman:FragmentTransfer>
```

```
(40)  </s:Header>
(41)  <s:Body>
(42)  <wsman:XmlFragment>
(43)   1
(44)  </wsman:XmlFragment>
(45)  </s:Body>
```

Если соответствующее обновление завершатся успешно, то новое представление соответствует запросу и в s:Body не ожидается результа, хотя возвращение его всегда правомерно. Если новове значение не соответствует запросу, служба должна предоставлять только те части, которые отличаются от соответствующих частей в запросе. Такая ситуация обычно не происходит для единичных значений, так как невозможность установить новое значение приводит к отказу wsmt:InvalidRepresentation.

Пример:

Ниже приведен типовой ответ:

```
(46) <s:Envelope
(47)    xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(48)    xmlns:wsa="gost.e-ecolog.ru
(49)    xmlns:wsman="gost.e-ecolog.ru                                    >
(50)   <s:Header>
(51)    <wsa:To>
(52)      gost.e-ecolog.ru
(53)    </wsa:To>
(54)    <wsa:Action s:mustUnderstand="true">
(55)     gost.e-ecolog.ru
(56)   </wsa:Action>
(57)   <wsa:MessageID s:mustUnderstand="true">
(58)    urn:uuid:ee7f13b5-0091-430b-9ed8-2e12fbaa8a7e
(59)   </wsa:MessageID>
(60)   <wsa:RelatesTo>
(61)    urn:uuid:d9726315-bc91-2222-9ed8-c044c9658a87
(62)   </wsa:RelatesTo>
(63)   <wsman:FragmentTransfer s:mustUnderstand="true">
(64)    BootPartition/text()
(65)   </wsman:FragmentTransfer>
(66)  </s:Header>
(67)  <s:Body>
(68)   <wsman:XmlFragment>
(69)    1
(70)   </wsman:XmlFragment>
(71)  </s:Body>
(72) </s:Envelope>
```

R7.9-1 Это правило преднамеренно оставлено незаполненным.

R7.9-2 Если служба сталкивается с попыткой обновить значение только для чтения при использовании операции Put на уровне фрагмента, она должна вернуть сообщение-отказ wsa:ActionNotSupported со следующим кодом детализации:

gost.e-ecolog.ru

Примечание — Операция Put на уровне фрагмента подразумевает замену или обновление и не вставляет новые значения в объект представления. Таким образом, нецелесообразно использовать Put, чтобы, например, вставить новое значение в конец массива. Можно возвратить целый и затем обновить и заменить (потому что таким образом происходит обновление целого массива), но единичная операция вставки нового элемента в середину или конец массива фактически является операцией Create.

Как указано в 7.4, если новое представление отличается от входных данных, тоновое представление должно быть возвращено в ответе. С Put на уровне фрагмента это правило применяется только к части обновляемого объекта представления, а не ко всему объекту. Если единственное значение записано и принято, но имеет побочные эффекты в представлении других значений, невозвращается весь объект.

Чтобы установить значение в NULL, не удаляя его как элемент, используйте значение атрибута xsi:nil на элементе, который нужно установить в NULL, чтобы гарантировать, что путь фрагмента построен соответствующим образом.

Пример:

```
(73)  <s:Header> ...
(74)   <wsman:FragmentTransfer s:mustUnderstand="true">
(75)    AssetLabel
(76)   </wsman:FragmentTransfer>
(77)   ...
(78)  </Header>
(79)  <s:Body>
(80)   <wsman:XmlFragment xmlns:xsi="gost.e-ecolog.ru                    >
(81)   <AssetLabel xsi:nil="true"/>
(82)  </wsman:XmlFragment>
(83)  </s:Body>
```

7.10 Delete на уровне фрагмента

Delete на уровне фрагмента применяется, только если XML-схема для предназначенного объекта поддерживает необязательные элементы, которые можно удалить из объекта представления, или массив (повторяющиеся элементы) с переменным числом элементов и клиент хочет удалить элемент из массива. Если необходима замена целого массива, может использоваться Put на уровне фрагмента. Для доступа к массиву может использоваться нотация доступа к массиву XPath. Чтобы удалить значение, которое правомерно удалять (согласно правилам схемы для объекта), выражение wsman:FragmentTransfer определяет объект для удаления.

Пример:

```
(1)  <wsman:FragmentTransfer s:mustUnderstand = "true">
(2)   VolumeLabel
(3)  </wsman:FragmentTransfer>
```

Для установки значения в пустой указатель (NULL), не удаляя его как элемент, используйте Put на уровне фрагмента со значением xsi:nil.

Для удаления элемента массива используются операторы XPath [].

Пример:

В следующем примере удаляется второй элемент <BlockedIPAddress> в представлении.

(Нумерация массивов XPath начинается с 1)

```
(1)  <wsman:FragmentTransfer s:mustUnderstand = "true">
(2)   BlockedIPAddress[2]
(3)  </wsman:FragmentTransfer>
```

<s:Body> пуст для всех операций Delete, даже с доступом на уровне фрагмента, и применяются все нормальные отказы для Delete.

R7.10-1 Если значение не может быть удалено из-за блокировки или аналогичных условий, следует вернуть сообщение-отказ wsman:AccessDenied.

7.11 Create на уровне фрагмента

Create на уровне фрагмента применяется только в случае, если XML-схема для предназначенных объектов поддерживает необязательные элементы, не присутствующие в настоящее время, или под-

держивает массивы с переменным числом элементов,= и клиент хочет вставить элемент в массив (повторяющийся элемент). Если необходима замена целого массива, может использоваться Put на уровне фрагмента. Для доступа к массиву может использоваться нотация доступа к массиву XPath (операторы[]).

Примечание — Create может использоваться только для добавления нового содержимого без обновления существующего содержимого.

Для вставки значения, которое можно правомерно добавить (согласно правилам схемы для объекта), выражение wsman:FragmentTransfer определяет элемент, который будет добавлен.

Пример:

Предположим, что следующий фрагмент сообщения отправляется в ресурс LogicalDisk:

(1) <wsman:FragmentTransfer s:mustUnderstand="true">

(2) VolumeLabel

(3) </wsman:FragmentTransfer>

Пример 2: В этом случае <Body> содержит и элемент, и значение:

(4) <s:Body>

(5) <wsman:XmlFragment>

(6) <VolumeLabel> MyDisk </VolumeLabel>

(7) </wsman:XmlFragment>

(8) </s:Body>

Эта операция создает элемент <VolumeLabel>в случае, когда VolumeLabel прежде не существовал.

Примеры:

1 Для создания с использованием только одного значения применяется оператор XPath text() к пути следующим образом:

(9) <wsman:FragmentTransfer s:mustUnderstand="true">

(10) VolumeLabel/text()

(11) </wsman:FragmentTransfer>

2 Тело Create содержит значение, которое будет вставлено, совпадает с Put на уровне фрагмента:

(12) <s:Body>

(13) <wsman:XmlFragment>

(14) MyDisk

(15) </wsman:XmlFragment>

(16) </s:Body>

Для создания элемента массива может использоваться оператор XPath [ ]. Для вставки нового элемента в конце массива пользователь должен знать число элементов в массиве для задания нового индекса.

Пример:

Следующий фрагмент сообщения направляется в ресурс InternetServer:

(17) <wsman:FragmentTransfer s:mustUnderstand="true">

(18) BlockedIPAddress[3]

(19) </wsman:FragmentTransfer>

Вставка нового элемента в пределах массива реализуется с использованием индекса желаемого местоположения, причем в этом местоположении массив расширяется, чтобы вместить новый элемент. Использование Put в этом местоположении переписывает существующий элемент массива, тогда как Create вставляет новый элемент, увеличивая массив.

Тело Create содержит значение, которое будет вставлено, и оно совпадает с Put на уровне фрагмента.

Пример:

```
(20)  <s:Body>
(21)  <wsman:XmlFragment>
(22)  <BlockedIPAddress> 123.12.188.44 </BlockedIPAddress>
(23)  </wsman:XmlFragment>
(24)  </s:Body>
```

Эта операция добавляет третий IP-адрес к массиву <BlockedIPAddress> (повторяющиеся элементы) в предположении, что по крайней мере два элемента уже находятся на том уровне.

R7.11-1 Служба не должна использовать Create на уровне фрагмента, чтобы изменить значение существующего свойства. Если целевой объект уже существует, следует вернуть сообщение-отказ wsman:AlreadyExists.

R7.11-2 Если Create не состоялось по причине того, что результат в некотором роде не соответствует схеме, следует вернуть сообщение-отказ wsmt:InvalidRepresentation.

Как определено в 7.6, CreateResponse содержит EPR созданного ресурса. В случае Create на уровне фрагмента ответ дополнительно содержит в заголовке блок wsman:FragmentTransfer, включающий путь (строка 12).

Пример:

В следующем примере ResourceCreated EPR продолжает обращаться ко всему объекту, а не только к фрагменту.

```
(25)  <s:Envelope
(26)    xmlns:s="gost.e-ecolog.ru                    03/05/soap-envelope"
(27)    xmlns:wsa="gost.e-ecolog.ru
(28)    xmlns:wsman="gost.e-ecolog.ru
(29)    xmlns:wsmt="gost.e-ecolog.ru                                        >
(30)   <s:Header>
(31)    ...
(32)    <wsa:Action>
(33)      gost.e-ecolog.ru
(34)   </wsa:Action>
(35)   <wsman:FragmentTransfer s:mustUnderstand="true">
(36)    Path To Fragment
(37)   </wsman:FragmentTransfer>
(38)   ...
(39)  </s:Header>
(40)  <s:Body>
(41)  <wsmt:ResourceCreated>
(42)  <wsa:Address> ... </wsa:Address>
(43)  <wsa:ReferenceParameters>
(44)  <wsman:SelectorSet>
(45) <wsman:Selector ...> ... </wsman:Selector>
(46) </wsman:SelectorSet>
(47) </wsa:ReferenceParameters>
(48) </wsmt:ResourceCreated>
(49) </s:Body>
(50) </s:Envelope>
```

Как было указано в 7.6, для сохранения совместимости с WSDL, в SOAP Body возвращается только EPR объекта, несмотря на другие варианты, рассматриваемые в 7.6.

8 Перечисление наборов данных

8.1 Общие положения

Данный раздел определяет ряд операций, которые могут использоваться для перебора специфичных для управления элементов набора данных или коллекции. WS-Management определяет и расширяет эти операции, как описано в данном разделе.

Есть многочисленные приложения, для которых простая метафора «один запрос — один ответ» недостаточна для передачи больших наборов данных по SOAP. Запросы, которые не вписываются в эту простую парадигму, включают потоковую передачу, обход, запрос и перечисление.

Данный пункт определяет простой протокол для перечисления, основывающийся на SOAP, который позволяет источнику данных предоставить потребителю абстракцию сессии, называемую контекстом перечисления, которая представляет логический курсор последовательности элементов данных. В таком случае потребитель может запросить информационные объекты элементов XML, используя данный контекст перечисления в одном или нескольких сообщениях SOAP.

Где-то должно сохраняться состояние процедуры перечисления. Это состояние может сохраняться между запросами источником данных или потребителем данных. Операции, определенные в данном разделе, позволяют источнику данных принимать решение для каждого запроса, какая сторона отвечает за поддержание этого состояния для следующего запроса.

В простейшей форме есть единственная операция Pull, которая позволяет источнику данных, в контексте определенного перечисления, производить последовательность элементов XML в теле сообщения SOAP. Каждая последующая операция Pull возвращает следующие N элементов совокупной последовательности.

Источник данных может обеспечивать специализированный механизм для начала нового перечисления. Например, источник данных, который обеспечивает доступ к базе данных SQL, может поддерживать операцию SELECT, которая выполняет запрос к базе данных и использует явный курсор базы данных для перебора кортежей, возвращенных запросом. Однако в целом для всех источников данных бывает проще поддерживать единственную стандартную операцию начала перечисления. В качестве такой операции данная спецификация определяет операцию Enumerate, которую могут реализовать источники данных для старта нового перечисления источника данных. Операция Enumerate используется для создания новых контекстов перечисления для последующего перебора/получения. Каждая операция Enumerate порождает отличный от предыдущих контекст перечисления, каждый со своим собственным логическим курсором/позицией.

Нужно подчеркнуть, что различные перечисления одного и того же источника данных могут привести к различным результатам; это может произойти даже для двух контекстов перечисления, созданных одновременно единственным потребителем, использующим идентичные запросы Enumerate. В целом потребитель перечисления не должен делать предположения о порядке или полноте перечисления; возвращенные элементы данных предоставляют выборку по источнику элементов данных, которые он хочет представить тому потребителю в то время и в том порядке, без гарантии, что каждый доступный элемент будет возвращен или что порядок, в котором элементы возвращены, имеет какое-либо семантическое значение (конечно, любой конкретный источник данных может предоставить весомые гарантии, если это требуется). В частности, нужно отметить, что сам факт перечисления содержимого источника данных может изменить это содержимое источника данных; например, очередь может быть представлена как источник данных таким образом, что элементы, которые возвращены в ответе Pull, удаляются из очереди.

Контексты перечисления представляют определенный порядок обхода информационных объектов XML. Операция Enumerate может использоваться для установки контекста перечисления источника данных. Операция Pull используется для получения информационных объектов от источника данных согласно определенному контексту перечисления. Операция Release используется для того, чтобы сообщить источнику данных, что потребитель прекращает пользоваться контекстом перечисления раньше, чем он завершит перечисление.

Контексты перечисления представлены как данные XML, непрозрачные для потребителя. Первоначально потребитель получает контекст перечисления от источника данных посредством операции Enumerate. В этом случае потребитель передает эти данные XML обратно к источнику данных в запро-

се Pull. Источник данных может произвольно возвратить обновленный контекст перечисления в ответе Pull; в этом случае новый контекст перечисления должен заменить старый у потребителя, и его необходимо передавать источнику данных во всех последующих запросах до тех пор, пока (и если) источник данных снова не возвратит обновленный контекст перечисления.

Потребители не должны повторно использовать старые контексты перечисления, которые были заменены источником данных. Использование замененного контекста перечисления в ответе Pull может привести к неопределенным результатам, включая игнорирование контекста или создание отказа.

После того, как будет возвращен последний элемент в последовательности или закончится контекст перечисления, этот контекст перечисления считается недействительным, а результат последующих операций, ссылающихся на этот контекст, считается не определенным.

Потребители могут отправить операцию Release для действительного контекста перечисления в любое время, это приводит к тому, что контекст перечисления становится недействительным, и позволяет источнику данных освободить любые ресурсы, которые он мог выделить для перечисления. Выполнение операции Release до окончания последовательности элементов явным образом разрешено; однако после операции Release никакие дальнейшие операции не следует выполнять.

Кроме того, при необходимости в любое время источник данных может отменить контекст перечисления.

Если ресурс с несколькими экземплярами обеспечивает механизм для перечисления или запроса набора экземпляров, операции, определенные в данном разделе, могут использоваться для такого перечисления.

R8.1-1 Служба может поддерживать операции Enumerate, если поддерживается перечисление какого-либо вида.

R8.1-2 Если через Веб-службы предоставляется простое нефильтрованное перечисление экземпляров ресурса, служба, соответствующая настоящему стандарту, должна поддерживать операции Enumerate для такого предоставления. Для перечисления экземпляров служба может поддерживать также и другие методы.

R8.1-3 Если фильтрованное перечисление (запросы) экземпляров ресурса предоставляется через Веб-службы, совместимой службе следует поддерживать операции Enumerate для такого перечисления. Для перечисления экземпляров служба может поддерживать также и другие методы.

Данный пункт указывает, что перечисление — это трехшаговая операция:

1) отсылается первоначальное сообщение Enumerate для установления контекста перечисления;

2) для итерирования набора результатов используются операции Pull;

3) когда итератор перечисления больше не требуется, но еще не исчерпан, отсылается сообщение Release, чтобы освободить счетчик и связанные ресурсы.

Как и в случае других методов WS-Management, перечисление может использовать wsman: OptionSet.

R8.1-4 Служба может отправить сообщения wsmen:Renew, wsmen:GetStatus и wsmen: EnumerationEnd; однако для ограниченных реализаций это кандидаты на исключение. Если эти сообщения не поддерживаются, то в ответ на эти запросы должен возвратиться отказ wsa:ActionNotSupported.

R8.1-5 Если служба предоставляет перечисление, то она должна, по крайней мере, поддерживать следующие сообщения: Enumerate, Pull и Release и связанные с ними ответы.

Если служба не поддерживает счетчики с состоянием, Release не будет иметь побочных эффектов и тривиальна для реализации. (Она всегда завершается успешно, если операция корректна.) Однако в этом случае она поддерживается для унификации конструирования клиентов.

R8.1-6 Операции Pull и Release являются продолжением первоначальной операции Enumerate. Службе следует обеспечивать ту же самую аутентификацию и авторизацию всюду по всей последовательности операций, а также следует возвращать ошибку для любой попытки изменить идентификационные данные во время последовательности.

Некоторые транспортные средства, такие как HTTP, могут сбросить или восстановить соединение между Enumerate и последующими операциями Pull либо между операциями Pull. Ожидается, что служба позволит продолжить перечисления без прерываний, но по практическим причинам некоторые службы могут потребовать, чтобы использовалось то же самое соединение. Данная спецификация не устанавливает требований в этом отношении. Однако R8.1-6 устанавливает, что идентификационные данные пользователя не изменяются на протяжении всей последовательности перечисления.

8.2 Перечисление (Enumerate)

Все источники данных должны поддерживать некоторую операцию, которая позволяет начать перечисление. Источник данных может поддерживать операцию Enumerate или может обеспечить иной механизм для начала перечисления и получения контекста перечисления.

Операция Enumerate инициируется после отправления сообщения-запроса Enumerate в источник данных. Сообщение-запрос Enumerate должно иметь следующую форму:

```
(1) <s:Envelope ...>
(2)   <s:Header ...>
(3)     <wsa:Action>
(4)       gost.e-ecolog.ru
(5)     </wsa:Action>
(6)     <wsa:MessageID>xs:anyURI</wsa:MessageID>
(7)     <wsa:To>xs:anyURI</wsa:To>
(8)     ...
(9)   </s:Header>
(10)  <s:Body ...>
(11)    <wsmen:Enumerate ...>
(12)      <wsmen:EndTo>ссылка-на-оконечную-точку</wsmen:EndTo> ?
(13)      <wsmen:Expires>[xs:dateTime | xs:duration]</wsmen:Expires> ?
(14)      <wsmen:Filter Dialect="xs:anyURI"?> xs:any </wsmen:Filter> ?
(15)      ...
(16)    </wsmen:Enumerate>
(17)  </s:Body>
(18) </s:Envelope>
```

Ниже описаны дополнительные нормативные ограничения на приведенную схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение:

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body / */wsmen:EndTo

Данный необязательный элемент указывает, куда отправить сообщение EnumerationEnd, если перечисление неожиданно закончилось. Данный элемент (если таковой присутствует) должен иметь тип wsa:EndpointReferenceType. По умолчанию это сообщение не отправляется. Конечная точка, на которую ссылается данный EPR, должна реализовать привязку «EnumEndEndpoint» вида portType, описанного в приложении З.

/s:Envelope/s:Body / */wsmen:Expires

Требуемое время срока окончания для перечисления. (Нет значения по умолчанию.) Источник данных определяет фактический срок окончания и может использовать время меньшее или большее, чем запрошенный срок окончания. Время срока окончания может быть определенным временем или продолжительностью (длительностью интервала времени с момента создания перечисления). И явно заданное время, и продолжительность интерпретируются по часам источника данных.

Если данный элемент отсутствует, то запрос соответствует перечислению, срок окончания которого не задан. Таким образом, потребитель направляет источнику данных запрос на создание перечисления с неопределенным периодом жизни. Если источник данных предоставит такое перечисление, оно завершится, когда будет достигнут конец перечисления, либо, если потребитель отправит запрос Release, либо его завершит источник данных в любое время по причине завершения соединения, ограничения ресурсов или завершения работы системы.

Если срок окончания будет с нулевой продолжительностью или конкретным временем, которое происходит в прошлом согласно источнику данных, то запрос не должен выполняться, и источник данных может вернуть отказ wsmen:InvalidExpirationTime, указывающий, что было затребовано недействительное время срока окончания.

Некоторые источники данных не могут иметь доступных «настенных часов», и, таким образом, в состоянии принять срок окончания только в формате продолжительности. Если такой источник получит запрос Enumerate, содержащий явное время срока окончания, то запрос не должен быть выполнен; в таком случае источник данных должен вернуть отказ wsmen:UnsupportedExpirationType, указывающий, что был затребован неподдерживаемый тип срока окончания.

/s:Envelope/s:Body/wsmen:Enumerate/wsmen:Filter

Данный необязательный элемент содержит булевский предикат, записанный на некотором диалекте (см. /s:Envelope/s:Body/*/wsmen:Filter/@Dialect), которому должны удовлетворять все интересующие элементы. Результирующий контекст перечисления не должен возвращать элементы, для которых вычисление предиката дает значение false. Если данный элемент отсутствует, то значение по умолчанию — выражение true(), обозначающее, что фильтрация нежелательна.

Если источник данных не поддерживает фильтрацию, то запрос должен быть отброшен, источник данных может отправить отказ wsmen:FilteringNotSupported SOAP.

Если источник данных поддерживает фильтрацию, но не может обработать требуемый диалект фильтра, запрос должен быть отброшен, и источник данных может отправить отказ wsmen:FilterDialect RequestedUnavailable SOAP.

Если источник данных поддерживает фильтрацию и требуемый диалект, но не может обработать содержимое фильтра, запрос должен быть отброшен, и источник данных может отправить отказ wsman:CannotProcessFilter SOAP.

/s:Envelope/s:Body/*/wsmen:Filter/@Dialect

Значение по умолчанию “gost.e-ecolog.ru

/s:Envelope/s:Body/*/wsmen:Filter/@Dialect = “gost.e-ecolog.ru

Значение/s:Envelope/s:Body/*/wsmen:Filter является предикатным выражением XPath [XPath 1.0] (PredicateExpr); контекст выражения:

- узел контекста: любой элемент XML, который может возвратиться как прямой дочерний элемент элемента Items;
- положение в контексте: 1;
- размер контекста: 1;
- привязки переменных контекста: Нет;
- библиотеки функций: Основная библиотека функции [XPath 1.0];
- объявления пространства имен: свойство [in-scope namespaces][Инфо-набор XML] элемента /s:Envelope/s:Body/*/wsmen:Filter.

Другие компоненты приведенной выше схемы сообщения далее не ограничиваются данной спецификацией.

После успешной обработки сообщения-запроса Enumerate источник данных, как ожидается, создаст контекст перечисления и вернет этот контекст в ответном сообщении Enumerate, которое должно придерживаться следующей формы:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:ReplyTo>ссылка-на-оконечную-точку</wsa:ReplyTo>

(7) <wsa:To>xs:anyURI</wsa:To>

(8) ...

(9) </s:Header>

(10) <s:Body ...>

(11) ***<wsmen:EnumerateResponse ...>***

(12) ***<wsmen:Expires>[xs:dateTime | xs:duration]</wsmen:Expires> ?***

(13) ***<wsmen:EnumerationContext>...</wsmen:EnumerationContext>***

(14) ***...***

(15) ***</wsmen:EnumerateResponse>***

(16) ***</s:Body>***

(17) ***</s:Envelope>***

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение:

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/*/wsmen:Expires

Срок окончания, установленный источником данных. Время срока окончания может быть или абсолютным временем или продолжительностью, но должно быть того же типа, что и срок окончания в запросе (если таковой имеется).

Если данный элемент отсутствует, то у перечисления не истечет время жизни. Таким образом, перечисление имеет неопределенный период жизни. Оно завершится, когда будет достигнут конец перечисления, либо если потребитель отправит запрос Release, либо его завершит источник данных в любое время по причине разрыва связи, ограничения ресурсов или завершения работы системы.

/s:Envelope/s:Body/wsmen:EnumerateResponse/wsmen:EnumerationContext

Обязательный элемент EnumerationContext содержит представление XML нового контекста перечисления. Потребитель обязан передавать данные XML в запросах Pull о данном контексте перечисления, до тех пор (и если), пока сообщение PullResponse не обновит контекст перечисления.

8.2.1 Общие положения

WS-Management определяет операцию Enumerate, как описано в данном пункте.

R8.2.1-1 Служба, соответствующая настоящему стандарту, может принимать сообщения wsmen:Enumerate с адресом EndTo; однако, если EnumerationEnd не поддерживается, служба может вернуть отказ wsman:UnsupportedFeature со следующим кодом детализации: gost.e-ecolog.ru wbem/wsman/1/wsman/faultDetail/ AddressingMode

R8.2.1-2 Служба, соответствующая настоящему стандарту, должна принимать сообщение Enumerate со сроком окончания Expires или вернуть отказ wsman:UnsupportedFeature и следующим кодом детализации: gost.e-ecolog.ru

R8.2.1-3 Элемент wsman:Filter (см. 8.3) в теле Enumerate должен быть либо простым текстом, либо составным элементом XML. Служба, соответствующая настоящему стандарту, не должна принимать смешанное содержимое из текста и элементов или нескольких дочерних элементов XML элемента wsman:Filter.

Несмотря на то, что в общем случае для Enumerate разрешается использование смешанного содержимого, для реализаций WS-Management это излишне сложно.

Общим диалектом фильтра служит XPath 1.0 (URI диалекта gost.e-ecolog.ru TR/1999/ RECxpath-19991116). Реализации с ограниченными ресурсами могут испытывать затруднения в предоставлении полной обработки XPath и все же желательно использовать подмножество синтаксиса XPath. В тех случаях, когда выражение фильтра принадлежит собственному подмножеству указанного диалекта, это правомерно, и может быть описано с использованием этого значения диалекта.

Никакое правило не налагает требования на использование XPath или любого подмножества в качестве диалекта фильтрации. Если диалект не определен, интерпретация по умолчанию подразумевает, что значение фильтра — XPath (как определено ранее в данном пункте).

R8.2.1-4 Служба, соответствующая настоящему стандарту, может не поддерживать весь синтаксис и вычислительную мощность указанного диалекта фильтра. Единственное требование заключа-

ется в том, чтобы указанный Filter был синтаксически корректным в рамках определения диалекта. Поэтому подмножества правомерны. Если указанный Filter превышает возможность службы, следует вернуть сообщение-отказ wsmen:CannotProcessFilter с некоторым текстом, указывающим на причины отказа.

Некоторые службы требуют применения фильтров, так как их пространство поиска столь большое, что простое перечисление бессмысленно или невозможно.

R8.2.1-5 Если требуется wsman:Filter, то служба, соответствующая настоящему стандарту, должна отклонить любой запрос без wsman:Filter, используя отказа wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

R8.2.1-6 Служба, соответствующая настоящему стандарту, может заблокировать, отклонить (используя отказы wsman:Concurrency), или разрешить другие параллельные операции на ресурсе во время перечисления, и может включить или исключить результаты таких операций как часть любого перечисления, которое еще происходит.

Настоящий стандарт не накладывает ограничений на поведение перечисления для случая, когда клиенты выполняют другие операции, такие как Create или Delete одновременно с перечислением. Служба может включать или исключать результаты этих операций в режиме реального времени, может произвести начальный моментальный снимок перечисления и выполнять запросы Pull дляэтого снимка или может блокировать доступ к другим операциям на время выполнения перечисления.

8.2.2 Опция перечисления Count

Чтобы дать клиентам оценку числа элементов в перечислении, определены два дополнительных заголовка SOAP: один для использования в сообщении-запросе для запроса приблизительного числа объектов в последовательности перечисления, а другой, соответствующий заголовок, — для использования в ответе для возвращения этого значения клиенту.

Эти заголовки SOAP определены для использования с сообщениями Enumerate и Pull и их ответов. Заголовки, используемые в Enumerate и Pull, следующие:

(1) <s:Header>

(2) ...

(3) <wsman:RequestTotalItemsCountEstimate ... />

(4) </s:Header>

Заголовок, используемый службами для возврата значения, следующий:

(1) <s:Header>

(2) ...

(3) <wsman:TotalItemsCountEstimate>

(4) xs:nonNegativeInteger

(5) </wsman: TotalItemsCountEstimate>

(6) </s:Header>

Ниже приведены дополнительные нормативные ограничения на приведенные выше заголовки:

wsman:RequestTotalItemsCountEstimate при наличии в сообщениях SOAP Enumerate или Pull указывает, что клиент просит включить в связанное ответное сообщение оценку общего количества элементов в последовательности перечисления.

Настоящий стандарт не определяет семантику этого заголовка SOAP в любых других сообщениях.

wsman:TotalItemsCountEstimate при наличии в сообщении SOAP EnumerateResponse или PullResponse указывает на приблизительное число объектов в последовательности перечисления.

Это общее число объектов, а не оставшееся число объектов в последовательности. Настоящий стандарт не определяет семантику этого заголовка SOAP в любых других сообщениях.

Когда служба понимает функцию TotalItemsCountEstimate, но не может определить число объектов, служба отвечает с помощью элемента wsman:TotalItemsCountEstimate, имеющего атрибут xsi:nil со значением «true», следующим образом:

(1) <wsman:TotalItemsCountEstimate xsi:nil = “true” />

R8.2.2-1 Служба, соответствующая настоящему стандарту, может поддерживать возможность возвратить оценку числа элементов в последовательности перечисления. Если служба получит со-

общение Enumerate или Pull без заголовка wsman:RequestTotalItemsCountEstimate SOAP, то служба не должна возвращать заголовок wsman:TotalItemsCountEstimate SOAP в связанном ответном сообщении.

R8.2.2-2 Значение, возвращенное в заголовке SOAP wsman:TotalItemsCountEstimate, является только оценкой числа элементовв последовательности. Клиенту не следует использовать значение wsman:TotalItemsCountEstimate для определения конца перечисления вместо использования EndOfSequence.

Данный механизм предназначен помочь клиентам в определении процента завершения перечисления во время выполнения. Когда служба отправляет оценку количества результатов после предшествующей оценки для той же самой последовательности перечисления, предполагается, что новая оценка полного количества результатов будет более точной.

8.2.3 Оптимизация перечислений с небольшими наборами результатов

Чтобы оптимизировать число пар запрос-ответ, требуемых для перечисления объектов в перечисляемом ресурсе, клиент может запросить оптимизированное поведение перечисления. Это поведение полезно в случаях, когда у перечисления есть такое небольшое число объектов, что начальный EnumerateResponse может обоснованно включить весь результат без потребности в последующем Pull для получения элементов. Данный механизм может использоваться даже для больших перечислений, чтобы получить первые несколько результатов в начальном ответе.

Клиент начинает оптимизированное перечисление, помещая wsman:OptimizeEnumeration элемент как дочерний элемент элемента Enumerate, а также может дополнительно включить элемент wsman:MaxElements следующим образом:

Пример:

```
(1) <s:Body>
(2)   <wsmen:Enumerate>
(3)     ...
(4)     <wsman:OptimizeEnumeration/>
(5)     <wsman:MaxElements>xs:positiveInteger</wsman:MaxElements> ?
(6)   </wsmen:Enumerate>
(7) </s:Body>
```

Ниже представлены дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsmen:Enumerate/wsman:OptimizeEnumeration, если размещен как дочерний элемент элемента Enumerate, то указывает, что клиент запрашивает оптимизированное перечисление;

wsmen:Enumerate/wsman:MaxElements,

(необязательный) указывает на максимальное число элементов, которые потребитель готов принять в EnumerateResponse.

Этот элемент играет ту же самую роль, что wsmen:Pull/wsmen:MaxElements. Когда данный элемент отсутствует, его значение по умолчанию равняется 1.

R8.2.3-1 Служба, соответствующая настоящему стандарту, может поддерживать оптимизацию перечисления. Если служба, не поддерживающая оптимизацию, получает элемент wsman:OptimizeEnumeration в сообщении Enumerate, следует проигнорировать элемент и завершить запрос перечисления как будто элемент не присутствовал.

Если служба игнорирует элемент, клиент продолжает с запросом Pull, как если не использовал опцию. Клиент не требует никаких специальных механизмов по тому, что было необходимо для нормального перечисления, если запрос на оптимизацию проигнорирован.

R8.2.3-2 Совместимая служба, которая получает сообщение Enumerate без элемента wsman:OptimizeEnumeration, не должна возвращать элементы перечисления в сообщение EnumerateResponse и должна вернуть инициализированный EnumerationContext, чтобы вернуть первые объекты, когда будет получено первое сообщение Pull.

Если служба осуществит оптимизацию, даже если это не требовалось, то клиенты, не знающие об оптимизации, неправильно обработают результат перечисления.

R8.2.3-3 Совместимая служба, которая получает сообщение Enumerate с элементом wsman:OptimizeEnumeration, не должна возвращать в ответном сообщении Enumerate больше элементов, чем требуется в элементе wsman:MaxElements (или более одного элемента, если элемент wsman:MaxElements не должен присутствовать). Реализации могут вернуть меньше элементов, осно-

вываясь на заголовке SOAP wsman:OperationTimeout, заголовке SOAP wsman:MaxEnvelopeSize или определенных ограничениях реализации.

После запроса клиента служба, осуществляющая оптимизированное перечисление, ответит следующим дополнительным содержимым в сообщении EnumerateResponse:

(1) <s:Body>
(2) <wsmen:EnumerateResponse>
(3) <wsmen:EnumerationContext> ... </wsmen:EnumerationContext>
(4) <wsman:Items>
(5) ...аналогично wsmen:Items в wsmen:PullResponse
(6) </wsman:Items> ?
(7) <wsman:EndOfSequence/> ?
(8) ...
(9) </wsmen:EnumerateResponse>
(10) </s:Body>

Ниже представлены дополнительные нормативные ограничения к представленной выше схеме сообщения:

wsman:Items — (необязательный) содержит один или несколько элементов, специфичных для запрошенного перечисления, которые были бы закодированы в элементе Items в PullResponse.

Служба возвратит в этом списке не больше элементов, чем wsman:MaxElements, если в сообщении запроса будет определен wsman:MaxElements, или один элемент, если wsman:MaxElements был опущен.

wsman:EndOfSequence — (необязательный) указывает, что в перечислении более нет доступных элементов, и весь результат (даже если есть нулевые элементы) содержится в элементе wsman:Items.

wsmen:EnumerationContext — требуемый контекст для запроса дополнительных элементов, если таковые имеются, в последующих сообщениях Pull.

Если присутствует wsman:EndOfSequence, EnumerationContext не может использоваться в последующем запросе Pull. Служба следует демонстрировать то же самое использование отказа, которое произошло бы, если бы EnumerationContext использовались в запросе Pull после того, как элемент EndOfSequence возник в PullResponse. Несмотря на то, что элемент EnumerationContext должен присутствовать, не требуется никакое значение; поэтому в случаях, когда присутствует элемент wsman:EndOfSequence, значение для EnumerationContext может быть пустым.

Пример:
(1) <s:Body>
(2) <wsmen:EnumerateResponse>
(3) <wsmen:EnumerationContext/>
(4) <wsman:Items>
(5) Объекты
(6) </wsman:Items>
(7) <wsman:EndOfSequence/>
(8) ...
(9) </wsmen:EnumerateResponse>
(10) </s:Body>

R8.2.3-4 Совместимая служба, которая поддерживает оптимизированное перечисление и отвечает сообщением EnumerateResponse, в ответе должна включать элемент wsman:Items, элемент wsman:EndOfSequence или оба, как признак для клиента, что оптимизированный запрос перечисления был понят и выполнен.

Если ни wsman:Items, ни wsman:EndOfSequence не находятся в сообщении EnumerateResponse, клиент может продолжать использовать обмены сообщениями перечисления, как определено в 8.2.1.

R8.2.3-5 Совместимая служба, которая поддерживает оптимизированное перечисление и не возвращает все элементы последовательности перечисления в сообщении EnumerateResponse, должна возвратить элемент EnumerationContext, который инициализирован таким образом, что последующее

сообщение Pull вернет набор элементов, следующие за возвращенными в EnumerateResponse. Если все элементы последовательности перечисления возвращены в сообщении EnumerateResponse, служба в ответе должна вернуть пустой элемент EnumerationContext и элемент wsman:EndOfSequence.

Клиент, который запросил оптимизированное перечисление, может определить, был ли данный запрос понят и принят службой, путем проверки ответного сообщения.

Клиенты, для которых важен размер начального ответа независимо от числа элементов, могут использовать механизм wsman:MaxEnvelopeSize, описанный в 6.2.

8.3 Интерпретация фильтра

Выражение элемента Filter может быть только булевским предикатом. Чтобы поддерживать произвольные специальные запросы, включая проекции, WS-Management определяет элемент wsman:Filter точно такой же формы, как в фильтре Enumerate за исключением того, что выражение фильтра Enumerate не обязано быть булевским предикатом. Это позволяет использовать перечисления с использованием существующих языков запросов, таких как SQL и CQL, которые объединяют предикат и информацию о проекциив одном синтаксисе. Использование проекции определено диалектом фильтра, а не спецификацией WS-Management.

(1) <wsman:FilterDialect="xs:anyURI"?>xs:any</wsman:Filter>

Атрибут Dialect является необязательным. Если атрибут не задан, подразумевается следующее значение:

gost.e-ecolog.ru ?/1999/REC-xpath-19991116

Данный диалект дает возможность использовать любое выражение полного XPath или подмножества.

Элемент wsman:Filter является дочерним элементом элемента Enumerate.

Если диалектом фильтра, используемого для сообщения Enumerate, служит XPath 1.0, узел контекста совпадает с тем, который определен в 8.1.

R8.3-1 Если служба поддерживает перечисление с фильтрацией, использующие Filter, то она должна также поддерживать фильтрацию с использованием wsman:Filter. Это правило позволяет стекам клиента всегда выбирать пространство имен XMLwsman для элемента Filter. Даже если служба поддерживает wsman:Filter, он не требуется для поддержания проекции.

R8.3-2 Если служба поддерживает перечисление с фильтрацией, использующее wsman:Filter, она должна также поддерживать фильтрацию с использованием Filter.

R8.3-3 Если запрос Enumerate содержит и Filter, и wsman:Filter, то служба должна вернуть сообщение-отказ wsmen:CannotProcessFilter.

Фильтры обычно предназначаются для выбора полных XML-документов представлений. Однако у большинства языков запросов есть и фильтрация, и композиционные возможности, в которых они могут возвратить подмножества оригинального представления или выполнить сложные операции на оригинальном представлении и возвратить что-то полностью новое.

Настоящий стандарт не устанавливает ограничений для возможностей службы, но службы могут предоставлять только простую возможность фильтрации, без композиционных возможностей. В целом фильтрующие диалекты подчиняются следующей простой иерархии:

1) простое перечисление без фильтрации;

2) фильтрованное перечисление без изменения представления (в пределах возможностей XPath, например);

3) фильтрованное перечисление, в котором отобрано подмножество каждого элемента (в пределах возможностей XPath, например);

4) композиция нового результата (XQuery), включая простую проекцию.

Большинство служб попадают в первую или вторую категорию. Однако, если служба хочет поддерживать перечисление на уровне фрагмента, чтобы обеспечить необязательный доступ на уровне фрагмента (7.7), служба также может реализовать категорию 3. Очень редко службы внедряют категорию 4.

XPath 1.0 может использоваться просто для фильтрации или может использоваться для того, чтобы, передать обратно подмножества представления (или даже значения, не обернутые в XML). В случаях, где результат не только фильтрован, но и «изменен», применяется техника 8.6.

Для случаев, когда XPath не может поддерживаться полностью, описано общее подмножество в приложение D.3.

Пример:

Ниже приведен типичный пример использования XPath в фильтре. Предположим, что у каждого элемента в перечислении, которое будет доставлено, есть следующее содержимое XML:

(1) <s:Body>

(2) ...

(3) <wsmen:Items>

(4) <DiskInfo xmlns="...">

(5) <LogicalDisk>C:</LogicalDisk>

(6) <CurrentMegabytes>12</CurrentMegabytes>

(7) <BackupDrive> true </BackupDrive>

(8) </DiskInfo>

(9) ...

(10) </wsmen:Items>

(11) </s:Body>

Для каждого объекта внутри обертки Items точка привязки для оценки XPath находится в первом элементе, и она не ссылается на элементы s:Body или Items. Выражение XPath оценивается, как если бы каждый элемент в блоке Items был отдельным документом.

Пример:

При использовании для простой обработки документов следующие четыре выражения XPath «выбирают» весь узел DiskInfo:

(12) /

(13) /DiskInfo (14)

(14) ../DiskInfo

(15) .

Если это выражение XPath используется в качестве «фильтра», оно не фильтрует экземпляры и совпадает с отбором всех экземпляров или с полным отсутствием фильтра. Однако, используя следующий синтаксис, выражение XPath выбирает узел XML, только если проверяемое выражение в скобках оценивается как логическое значение «true»:

(1) ../DiskInfo [LogicalDisk = «C»:]

В этом случае элемент будет отобран, только если он относится к дисководу «C»:; в противном случае узел XML не будет отобран. Это выражение XPath отфильтровывает все экземпляры DiskInfo для других драйверов.

Пример:

Полные реализации XPath могут поддерживать более сложные проверяемые выражения следующим образом:

(1) ../DiskInfo [CurrentMegabytes>"10" andCurrentMegabytes<"200"]

Это действие выбирает только диски со свободным пространством в пределах диапазона определенных значений.

В сущности XML-форма логически пропускается через процессор XPath, чтобы проверить, будет ли она отобрана. Если да, то она поступает в перечисление. В противном случае, от элемента отказываются и не поставляют как часть перечисления.

См. связанный подпункт 10.2.2 про фильтрации по подпискам.

8.4 Pull

Операция Pull начинается при отправлении сообщения-запроса Pull в источник данных. Сообщение-запрос Pull должно иметь следующую форму:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4)

gost.e-ecolog.ru

(5) ***</wsa:Action>***

(6) ***<wsa:MessageID>xs:anyURI</wsa:MessageID>***

(7) ***<wsa:ReplyTo>wsa:EndpointReference</wsa:ReplyTo>***

(8) ***<wsa:To>xs:anyURI</wsa:To>***

(9) ***...***

(10) ***</s:Header>***

(11) ***<s:Body ...>***

(12) ***<wsmen:Pull ...>***

(13) ***<wsmen:EnumerationContext>...</wsmen:EnumerationContext>***

(14) ***<wsmen:MaxTime>xs:duration</wsmen:MaxTime> ?***

(15) ***<wsmen:MaxElements>xs:long</wsmen:MaxElements> ?***

(16) ***<wsmen:MaxCharacters>xs:long</wsmen:MaxCharacters> ?***

(17) ***...***

(18) ***</wsmen:Pull>***

(19) ***</s:Body>***

(20) ***</s:Envelope>***

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение:

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/wsmen:Pull/wsmen:EnumerationContext

Данный обязательный элемент содержит XML-данные, которые представляют текущий контекст перечисления.

Если контекст перечисления недействителен из-за того, что он ранее был заменен в ответе на другой запрос Pull, он завершен (EndOfSequence был возвращаен в ответе на Pull), он был отменен (Release), он истек либо источник данных, был вынужден отменить контекст, то источнику данных следует отклонить запрос, и он может отправить отказ wsmen:InvalidEnumerationContext.

Источник данных может оказаться неспособен определить, что контекст перечисления недействителен, особенно если все состояния, связанные с перечислением, сохранены в контексте перечисления и обновляются в каждом PullResponse.

/s:Envelope/s:Body/wsmen:Pull/wsmen:MaxTime

Данный необязательный элемент (типа xs:duration) указывает на максимальное количество времени, в течение которого инициатор готов позволить источнику данных составлять ответ на Pull. При отстутствии данного элемента от источника данных не требуется ограничивать количество времени для составления ответа Pull.

Это применимо для источников данных, которые накапливают элементы в течение долгого времени и упаковывают их в единственный ответ Pull.

/s:Envelope/s:Body/wsmen:Pull/wsmen:MaxElements

Данный необязательный элемент (типа xs:long) указывает на число элементов (дочерних элементов элемента Items в ответе на Pull), которое готов принять потребитель. При отсутствии данного элемента его значение по умолчаню равно 1. Реализации не должны возвращать больше этого числа элементов в ответном сообщении Pull. Реализации могут возвратить меньше этого числа, основываясь на ограничениях времени выполнения операции MaxTime, предела размера MaxCharacters или специфичных для реализации ограничениях.

/s:Envelope/s:Body/wsmen:Pull/wsmen:MaxCharacters

Данный необязательный элемент (типа xs:long) указывает на максимальный размер возвращенных элементов в знаках Unicode, которые готов принять инициатор. При отсутствии данного элемента от источника данных не требуется ограничивать число знаков в ответе на Pull. Реализации не должны возвращать ответное сообщение Pull, у которого размер элемента Items превышает MaxCharacters. Реализации могут возвратить меньшее сообщение, основываясь на ограничениях времени выполнения операции MaxTime, пределе MaxElements или специфичных для реализации ограничениях.

Даже если запрос Pull будет содержать элемент MaxCharacters, потребитель должен быть готов получить ответ Pull, который содержит больше символов данных, чем указанно, так как алгоритмы каноникализации или альтернативной сериализации XML могут изменить размер представления.

Может случиться так, что следующий элемент, который источник данных должен возвратить потребителю, будет больше, чем MaxCharacters. В этом случае источник данных может пропустить элемент или может возвратить сокращенное представление элемента, которое умещается внутри MaxCharacters. Если источник данных пропускает элемент, он может вернуть его как часть ответа на будущий запрос Pull с большим значением MaxCharacters или опустить его полностью, исключив из перечисления. Если слишком большой элемент окажется последним элементом, который будет возвращен для данного контекста перечисления, и источник данных пропускает его, то источник данных должен включить элемент EndOfSequence в ответе Pull и отменить контекст перечисления; то есть, он может не возвратить нулевые элементы, и при этом не считать перечисление законченным. Посмотрите ниже в данном пункте обсуждение EndOfSequence.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

По получении сообщения-запроса Pull источник данных может ждать, сколько он сочтет необходимым (но не дольше, чем значение элемента MaxTime, если присутствует), чтобы создать сообщение для доставки потребителю. Источник данных должен признать элемент MaxTime и вернуть сообщение-отказ wsmen:TimedOut, если не будут доступны какие-либо элементы до срока окончания из сообщения-запроса.

Однако данный отказ не должен быть причиной того, что контекст перечисления станет недействительным (конечно, источник данных может отменить контекст перечисления по другим причинам). Таким образом, отправитель запроса должен быть способен отправить дополнительные запросы Pull, используя данный контекст перечисления, после получения такого отказа.

После успешной обработки сообщения-запроса Pull источник данных, как ожидается, вернет ответное сообщение Pull, которое должно придерживаться следующей формы:

```
(1)  <s:Envelope ...>
(2)    <s:Header ...>
(3)    <wsa:Action>
(4)    gost.e-ecolog.ru
(5)    </wsa:Action>
(6)    <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
(7)    <wsa:To>xs:anyURI</wsa:To>
(8)     ...
(9)    </s:Header>
(10)   <s:Body ...>
(11)   <wsmen:PullResponse ...>
(12)   <wsmen:EnumerationContext>...</wsmen:EnumerationContext> ?
(13) <wsmen:Items> ?
(14)    <xs:any> элемент специфичный-для-перечисления</xs:any> +
(15)   </wsmen:Items>
(16)   <wsmen:EndOfSequence/> ?
(17)    ...
(18)   </wsmen:PullResponse>
(19)   </s:Body>
(20) </s:Envelope>
```

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение:

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/wsmen:PullResponse/wsmen:EnumerationContext

Если присутствует необязательный элемент EnumerationContext, он содержит новое представление XML текущего контекста перечисления. Потребитель обязан заменить предшествующее представление содержимым этого элемента.

/s:Envelope/s:Body/wsmen:PullResponse/wsmen:Items/any

Необязательный элемент Items содержит один или несколько элементов, специфичных для перечисления, по одному для каждого возвращаемого элемента.

/s:Envelope/s:Body/wsmen:PullResponse/wsmen:EndOfSequence

Данный необязательный элемент указывает на то, что больше нет доступных элементов для перечисления. Кроме того, как только данный элемент возвращается в ответном сообщении Pull, на последующие запросы Pull с использованием этого контекст перечисления следует возвращать сообщение-отказ InvalidEnumerationContext; в любом случае на них не должен возвращаться действительный PullResponse.

По крайней мере, должен появиться один из Items или EndOfSequence либо, возможно, оба, если элементы возвращены и последовательность исчерпана. Подобным образом, EnumerationContext и EndOfSequence не должны присутствовать одновременно; оба могут отсутствовать либо присутствует один без другого, но не оба в одном и том же PullResponse.

Потребителю не следует высылать дополнительные сообщения-запросы Pull после того, как возвращен ответ Pull, содержащий элемент EndOfSequence. Таким образом, при получении ответа Pull, содержащего элемент EndOfSequence, потребителю не следует вызывать операцию Release, чтобы сигнализировать, что контекст перечисления больше не нужен.

Если потребитель отправляет Pull или Release с недействительным контекстом перечисления, результат не определен: источник данных может проигнорировать запрос или может вернуть сообщение-отказ InvalidEnumerationContext, как описано ранее в данном пункте, либо может принять другие меры.

Поскольку Pull позволяет клиенту определять широкий диапазон параметров группирования и времени, клиенту рекомендуется заранее узнать корректные диапазоны. Эта информация может быть экспортирована из службы в форме метаданных, которые выходят за рамки настоящего стандарта. В настоящем стандарте не определены какие-либо обмены сообщениями для обнаружения корректных диапазонов параметров.

Поскольку wsman:MaxEnvelopeSize можно запрашивать для любого ответа в WS-Management, это используется в сообщении Pull вместо MaxCharacters, который вообще является избыточным, и предпочтительнее его опускать. Однако, если wsman:MaxEnvelopeSize присутствует, он обладает следующими характеристиками:

R8.4-1 Если служба предоставляет операции по перечислению и поддерживает Pull с элементом MaxCharacters, службе следует реализовать MaxCharacters как общее руководство или подсказку, но она может проигнорировать его, если присутствует wsman:MaxEnvelopeSize, поскольку последний имеет более высокий приоритет. Служба не должна отбрасывать запрос в случае конфликта, но должна использовать значение wsman:MaxEnvelopeSize.

R8.4-2 Если служба предоставляет операции по перечислению и поддерживает Pull с элементом MaxCharacters, и единственный элемент ответа приводит к превышению ограничения, служба может возвратить единственный элемент с нарушением рекомендованного размера. Однако в любом случае служба не должна нарушать wsman:MaxEnvelopeSize.

Служба может отправить PullResponse с меньшим количеством элементов, чтобы гарантировать, что wsman:MaxEnvelopeSize не превышен. Однако, если единственный элемент может привести к его превышению, следует применять правила 6.2.

В общем случае MaxCharacters является рекомендацией, а wsman:MaxEnvelopeSize — строгим правилом.

R8.4-3 Если какой-либо отказ происходит во время Pull, совместимой службе следует позволить клиенту повторить Pull с другими параметрами, такими как большее значение ограничения или без ограничения, и попытаться получить элементы. Службе не следует отменять перечисление в целом, но следует сохранить достаточно контекста, чтобы суметь повторить его по желанию клиента. Однако, служба может отменить перечисление напрямую, если происходит ошибка с отказом InvalidEnumerationContext.

Если при запросе Pull происходит отказ, службе обычно не нужно отменять все перечисления, но можно просто остановить курсор и позволить клиенту попробовать еще раз.

Предполагается, что действительным является только EnumerationContext последнего ответа. Хотя служба может возвращать одни и те же значения EnumerationContext с каждым Pull, это не требуется, и фактически служба может изменить EnumerationContext непредсказуемым образом.

R8.4-4 Служба, соответствующая настоящему стандарту, может проигнорировать MaxTime, если также определен wsman:OperationTimeout, поскольку wsman:OperationTimeout имеет больший приоритет. Эти элементы обладают одной и той же семантикой и могут использоваться вместо друг друга. Если они используются вместе, службе следует использовать только элемент wsman:OperationTimeout.

Для унификации конструирования сообщений клиенты могут использовать wsman:OperationTimeout и wsman:MaxEnvelopeSize, но не MaxTime и MaxCharacters.

Любой отказ для Pull относится к самому сообщению Pull, а не связанному с ним перечислению. Новый EnumerationContext все еще считается действительным, и, если служба разрешает повторную попытку нового сообщения Pull, клиент может продолжить. Однако служба может досрочно прекратить перечисление при столкновении с любым видом проблемы (как определено в R8.4-7).

R8.4-5 Это правило преднамеренно оставлено незаполненным.

Если содержимое не доступно, перечисление все еще считают активным, и сообщение Pull может быть повторено.

R8.4-6 Если служба не может заполнить PullResponse никакими элементами до истечения срока окончания, следует вернуть сообщение-отказ wsman:TimedOut, чтобы указать, что действительно истек срок ожидания и что клиент вряд ли преуспеет, просто отправив другое сообщение Pull. Если служба ждет результатыв момент срока окончания, следует возвратить ответ без элементов и с обновленным EnumerationContext, который, возможно, изменился даже при том, что никакие элементы не были возвращены, следующим образом:

(1) <s:Body>

(2) <wsmen:PullResponse>

(3) <wsmen:EnumerationContext> ...возможно, обновленный...

</wsmen:EnumerationContext>

(4) <wsmen:Items/>

(5) </wsmen:PullResponse>

(6) </s:Body>

По существу, пустой блок объектов является командой от службы попробовать еще раз. Если служба отправляет сообщение об ошибке wsman:TimedOut, она подразумевает, что повторная попытка вряд ли обернется успехом. Как правило, служба знает, какое сообщение вернуть, на основании своего внутреннего состояния. Например, в самом первом сообщении Pull в случае, если служба ждет другой компонент, есть вероятность отказа wsman:TimedOut. Если перечисление продолжается без проблем и после 50 запросов истекает время ожидания очередного сообщения Pull, служба может просто передать обратно нулевые объекты, ожидая, что клиент может продолжить с другим сообщением Pull.

R8.4-7 Служба может досрочно завершить все перечисление в любое время, в этом случае возвращается отказ InvalidEnumerationContext. Любые дальнейшие операции невозможны, включая Release. В отдельных случаях, таких как внутренние ошибки или слишком большие ответы, могут быть возвращены другие отказы. Во всех таких случаях службе также следует отменить контекст перечисления.

R8.4-8 Если в сообщении PullResponse присутствует маркер EndOfSequence, элемент EnumerationContext должен быть опущен, поскольку перечисление завершено. Клиент не может впоследствии отправить сообщение Release.

Обычно о завершении перечисления во всех случаях сообщает элемент EndOfSequence, присутствующий в PullResponse, а не отказы. Если клиент пытается продолжить перечисление после его завершения, возвращается отказ InvalidEnumerationContext. Клиент не должен высылать сообщение Release, если произошло событие EndOfSequence, поскольку в таком случае перечисление завершено, и контекст перечисления недействителен.

R8.4-9 Если не указан элемент MaxElements, размер группы объектов равен 1.

R8.4-10 Если значение MaxElements больше, чем поддерживает служба, служба может проигнорировать значение и использовать любой собственный максимум по умолчанию.

Служба может экспортировать свое максимальное значение MaxElements в метаданных, но формат и местоположение таких метаданных выходят за рамки настоящего стандарта.

R8.4-11 Элемент EnumerationContext должен присутствовать во всех запросах Pull, даже если служба использует постоянную величину на время существования последовательности перечисления.

8.5 Release

Операция Release начинается отправкой сообщения-запроса Release в источник данных. Сообщение-запрос Release должно иметь следующую форму:

(1) ***<s:Envelope ...>***

(2) ***<s:Header ...>***

(3) ***<wsa:Action>***

(4) gost.e-ecolog.ru

(5) ***</wsa:Action>***

(6) ***<wsa:MessageID>xs:anyURI</wsa:MessageID>***

(7) ***<wsa:ReplyTo>wsa:EndpointReference</wsa:ReplyTo>***

(8) ***<wsa:To>xs:anyURI</wsa:To>***

(9) ***...***

(10) ***</s:Header>***

(11) ***<s:Body ...>***

(12) ***<wsmen:Release ...>***

(13) ***<wsmen:EnumerationContext>...</wsmen:EnumerationContext>***

(14) ***...***

(15) ***</wsmen:Release>***

(16) ***</s:Body>***

(17) ***</s:Envelope>***

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение:

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

/s:Envelope/s:Body/wsmen:Release/wsmen:EnumerationContext

Данный обязательный элемент содержит данные XML, которые представляют отменяемый контекст перечисления.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

После успешной обработки сообщения-запроса Release ожидается, что источник данных вернет ответное сообщение Release, которое должно придерживаться следующей формы:

(1) ***<s:Envelope ...>***

(2) ***<s:Header ...>***

(3) ***<wsa:Action>***

(4) gost.e-ecolog.ru

(5) ***</wsa:Action>***

(6) ***<wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>***

(7) ***<wsa:To>xs:anyURI</wsa:To>***

(8) ***...***

(9) </s:Header>

(10) <s:Body />

(11) </s:Envelope>

Ниже приведены дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Данный обязательный элемент должен содержать значение:

gost.e-ecolog.ru

Если в нижележащем транспорте также присутствует URI действия SOAP, его значение должно передавать то же самое значение.

Release используется только для того, чтобы выполнить досрочную отмену перечисления. В случаях, когда это не является фактически необходимым, реализация может предоставить фиктивную реализацию, которая всегда будет успешной. Это способствует унификации передачи сообщений со стороны клиента.

R8.5-1 Служба должна распознать и обработать сообщение Release, если перечисление будет досрочно завершено. Если маркер EndOfSequence присутствует в сообщении PullResponse, перечисление уже завершено, и сообщение Releasea не может быть выслано, поскольку не существует актуальный контекст перечисления.

R8.5-2 Клиент может не передавать сообщение Release своевременно либо может никогда не отправлять его. Служба, соответствующая настоящему стандарту, может завершить перечисление по истечении соответствующего времени простоя, и любая попытка снова использовать контекст перечисления должна привести к отказу InvalidEnumerationContext.

R8.5-3 Это правило преднамеренно оставлено незаполненным.

R8.5-4 Служба может получить сообщение Release асинхронно к любым уже выполняющимся запросам Pull и отменить перечисление. Служба может отказаться от такого асинхронного запроса и вернуть сообщение об ошибке wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Служба может также поставить в очередь или заблокировать запрос и сериализовать его таким образом, чтобы он был обработан после сообщения Pull.

В большинстве случаев службе желательно уметь асинхронно отменять невыполненное сообщение Pull. Эта возможность требует, чтобы служба могла получать сообщение Release асинхронно при обработке идущего сообщения Pull. Далее, она требует, чтобы информация, содержащаяся в элементе EnumerationContext, не изменялась между операциями Pull.

Примечание — Если значение EnumerationContext — просто возрастающее целое число, Release всегда использует предыдущее значение, таким образом, служба может считать его недействительным. Если элемент EnumerationContext содержит значение, которое не изменяется в запросах Pull (а также любую другую информацию, которая, возможно, нужна службе), службе может быть проще произвести отмену.

8.6 Специальные запросы и перечисления на уровне фрагмента

Как уже обсуждалось в 7.7, желательно, чтобы клиенты могли иметь доступ к подмножествам представления. Это особенно важно в случаях обработки информационных запросов, где пользователи обычно желают выполнить операции XPath или XQuery на представлении, для получения специальных результатов.

Поскольку сообщения SOAP должны соответствовать известным схемам, а информационные запросы возвращают динамически генерируемые результаты, которые могут не соответствовать ни одной схеме, для формирования ответов используется обертка wsman:XmlFragment (см. 7.7).

R8.6-1 Служба может поддерживать произвольные композиционные запросы, проекции или перечисления фрагментов объектов представления, предоставляя подходящий диалект в wsman:Filter. Получающийся набор Items в элементе PullResponse (или элементе EnumerateResponse, если используется OptimizedEnumeration) должен быть обрамлен элементами wsman:XmlFragment, как показано ниже:

(1) <s:Body>

(2) <wsmen:PullResponse>

(3) <wsmen:EnumerationContext> ..возможно, обновленный..</wsmen:EnumerationContext>
(4) <wsmen:Items>
(5) <wsman:XmlFragment>
(6) XML-содержимое
(7) </wsman:XmlFragment>
(8) <wsman:XmlFragment>
(9) XML-содержимое
(10) </wsman:XmlFragment>
(11) ...
(12) </wsmen:Items>
(13) </wsmen:PullResponse>
(14) </s:Body>

Схема для wsman:XmlFragment содержит директиву для подавления валидации схемы, разрешая валидирующему синтаксическому анализатору принять произвольное содержимое, произведенное процессором запросов, действующим за перечислением.

XPath 1.0 и XQuery 1.0 уже поддерживают возвращение подмножеств или композиции представлений таким образом, что те подходят для использования в этом отношении.

R8.6-2 Если служба не поддерживает перечисление на уровне фрагмента, следует возвратить отказ wsmen:FilterDialectRequestedUnavailable, тот же самый, что для любого другого неподдержанного диалекта.

Для фильтрации используется такое же выражение XPath, как описано в пунктах Enumeration (см. 8.2, 8.2.2, 8.2.3). Обертки wsman:XmlFragment применяются после оценки XPath для предотвращения нарушения схемы, если XPath выбирает наборы узлов, которые являются фрагментами и неправомерны относительно первоначальной схемы.

8.7 Перечисление EPR

Как правило, вывод EPR перчисляемых объектов просто путем инспекции невозможен. Во многих случаях желательно перечислить не сами объекты, а EPR этих объектов. Например, такие EPR могут применяться в последующих запросах Get или Delete. Подобным образом часто желательно перечислять как объекты, так и связанные с ними EPR.

Поведение по умолчанию для Enumerate определено в 8.1. Однако WS-Management предоставляет дополнительное расширение для управления результатом перечисления.

R8.7-1 Служба может поддерживать необязательный элемент модификатора wsman:EnumerationMode с значением *EnumerateEPR*, который возвращает как результат перечисления только EPR объектов.

Примеры:
1 (1) <s:Envelope ...>
(2) <s:Header>
(3) ...
(4) <wsa:Action>
(5) gost.e-ecolog.ru
(6) </wsa:Action>
(7) ...
(8) </s:Header>
(9) <s:Body>
(10) <wsmen:Enumerate>
(11) <wsman:Filter Dialect="...">фильтр</wsman:Filter>
(12) <wsman:EnumerationMode> EnumerateEPR </wsman:EnumerationMode>
(13) ...
(14) </wsmen:Enumerate>
(15) </s:Body>

(16) </s:Envelope>

2 В следующем примере показан гипотетический ответ на запрос из предыдущего примера:

(17) <s:Body>

(18) <wsmen:PullResponse>

(19) <wsmen:Items>

(20) <wsa:EndpointReference> ... </wsa:EndpointReference>

(21) <wsa:EndpointReference> ... </wsa:EndpointReference>

(22) <wsa:EndpointReference> ... </wsa:EndpointReference>

(23) ...

(24) </wsmen:Items>

(25) </wsmen:PullResponse>

(26) </s:Body>

Фильтр, если таковой присутствует, должен по-прежнему применяться к перечислению, но ответ содержит только EPR тех элементов, которые будут возвращены. Эти EPR предназначены для использования в последующих операциях Get.

R8.7-2 Служба может поддерживать необязательный модификатор wsman:EnumerationMode со значением *EnumerateObjectAndEPR*. Перечисленные объекты, если таковые существуют, оформлены в элемент wsman:Item, который сочетает два представления XML: представление данных, за которым следует связанное wsa:EndpointReference.

Пример 3:

Пример wsman:EnumerationMode выглядит следующим образом:

(1) <s:Header>

(2) ...

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) </s:Header>

(7) <s:Body>

(8) <wsmen:Enumerate>

(9) <wsman:Filter Dialect="...">фильтр</wsman:Filter>

(10) <wsman:EnumerationMode> EnumerateObjectAndEPR</wsman:EnumerationMode>

(11) ...

(12) </wsmen:Enumerate>

(13) </s:Body>

4: ответ выглядит следующим образом:

(1) <s:Body>

(2) <wsmen:PullResponse>

(3) <wsmen:Items>

(4) <wsman:Item>

(5) <PayloadObject xmlns="..."> ... </PayloadObject> <!-- Object -->

(6) <wsa:EndpointReference> ... </wsa:EndpointReference> <!-- EPR -->

(7) </wsman:Item>

(8) <wsman:Item>

(9) <PayloadObject xmlns="..."> ... </PayloadObject> <!-- Object -->

(10) <wsa:EndpointReference> ... </wsa:EndpointReference> <!-- EPR -->

(11) </wsman:Item>

(12) ...

(13) </wsmen:Items>

(14) </wsmen:PullResponse>

(15) </s:Body>

В предыдущем примере каждый элемент оформлен в обертку wsman:Item (строка 8), которая сама содержит объект представления (строка 9), и сопровождающий ее EPR (строка 10). Может присутствовать столько объектов wsman:Item, сколько совместимо с другими ограничениями кодирования.

R8.7-3 Если служба не поддерживает модификатор wsman:EnumerationMode, то она должна вернуть сообщение-отказ wsman:UnsupportedFeature со следующим кодом детализации: gost.e-ecolog.ru
dmtf.org/wbem/wsman/1/wsman/faultDetail/EnumerationMode.

8.8 Renew

Чтобы возобновить перечисление, потребитель отправляет источнику данных запрос следующей формы:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:MessageID>xs:anyURI</wsa:MessageID>

(7) <wsa:FaultTo>ссылка-на-оконечную-точку</wsa:FaultTo> ?

(8) <wsa:ReplyTo> ссылка-на-оконечную-точку </wsa:ReplyTo>

(9) <wsa:To>xs:anyURI</wsa:To>

(10) ...

(11) </s:Header>

(12) <s:Body ...>

(13) <wsmen:Renew ...>

(14) <wsmen:EnumerationContext>...</wsmen:EnumerationContext>

(15) <wsmen:Expires>[xs:dateTime | xs:duration]</wsmen:Expires> ?

(16) ...

(17) </wsmen:Renew>

(18) </s:Body>

(19) </s:Envelope>

Компоненты приведенной выше схемы сообщения ограничены, помимо ограничений на запрос создания перечисления, следующим дополнением (ями):

/s:Envelope/s:Body/*/wsmen:EnumerationContext

Данный обязательный элемент содержит данные XML, которые представляют текущий контекст перечисления.

Контекст перечисления может быть недействителен по причине его замены в ответе на другой запрос Pull или завершения (EndOfSequence был возвращен в ответе Pull), по причине его отмены потребителем или истечения срока, или по причине того, что источник данных вынужден был отменить контекст. Тогда источнику данных следует отклонить запрос, и он может отправить отказ wsmen:Invalid EnumerationContext.

Источник данных, возможно, не сумеет определить, что контекст перечисления недействителен, особенно если все состояния, связанные с перечислением, сохранены в контексте перечисления и обновляются в каждом PullResponse.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если источник данных принимает запрос возобновить перечисление, то он должен дать ответ следующей формы:

(1) <s:Envelope ...>

(2) <s:Header ...>

(3) <wsa:Action>

(4) gost.e-ecolog.ru

(5) </wsa:Action>

(6) <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
(7) <wsa:To>xs:anyURI</wsa:To>
(8) ...
(9) </s:Header>
(10) <s:Body ...>
(11) <wsmen:RenewResponse ...>
(12) <wsmen:Expires>[xs:dateTime | xs:duration]</wsmen:Expires> ?
(13) <wsmen:EnumerationContext>...</wsmen:EnumerationContext> ?
(14) ...
(15) </wsmen:RenewResponse>
(16) </s:Body>
(17) </s:Envelope>

Перечисленные компоненты приведенной выше схемы сообщения ограничены, помимо ограничений на ответ на запрос Enumerate, следующим дополнением:

/s:Envelope/s:Body/wsmen:RenewResponse/wsmen:Expires

Если запрашиваемый срок окончания представляет собой продолжительность, то подразумевается, что срок окончания будет отсчитываться от момента начала обработки запроса Renew источником данных.

/s:Envelope/s:Body/wsmen:RenewResponse/wsmen:EnumerationContext

В рассматриваемом ответе данный элемент является необязательным.

Если источник данных примет решение не возобновлять перечисление, то запрос должен быть отброшен, и источнику данных следует отправить отказ wsmen:UnableToRenew, указывающий, что возобновление не было принято.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

8.9 GetStatus

Чтобы получить статус перечисления, подписчик отправляет источнику данных запрос следующей формы:

(1) <s:Envelope ...>
(2) <s:Header ...>
(3) <wsa:Action>
(4) gost.e-ecolog.ru
(5) </wsa:Action>
(6) <wsa:MessageID>xs:anyURI</wsa:MessageID>
(7) <wsa:FaultTo>ссылка-на-оконечную-точку</wsa:FaultTo> ?
(8) <wsa:ReplyTo>ссылка-на-оконечную-точку</wsa:ReplyTo>
(9) <wsa:To>xs:anyURI</wsa:To>
(10) ...
(11) </s:Header>
(12) <s:Body ...>
(13) <wsmen:GetStatus ...>
(14) <wsmen:EnumerationContext>...</wsmen:EnumerationContext> ?
(15) ...
(16) </wsmen:GetStatus>
(17) </s:Body>
(18) </s:Envelope>

На компоненты приведенной выше схемы сообщения накладываются те же ограничения, что и на запрос возобновить перечисление. Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если перечисление действительно, и его срок окончания не наступил, то источник данных должен дать ответ следующей формы:

```
(1) <s:Envelope ...>
(2) <s:Header ...>
(3) <wsa:Action>
(4) gost.e-ecolog.ru
(5) </wsa:Action>
(6) <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
(7)   <wsa:To>xs:anyURI</wsa:To>
(8)   ...
(9)  </s:Header>
(10) <s:Body ...>
(11) <wsmen:GetStatusResponse ...>
(12) <wsmen:Expires>[xs:dateTime | xs:duration]</wsmen:Expires> ?
(13)     ...
(14)  </wsmen:GetStatusResponse>
(15)  </s:Body>
(16) </s:Envelope>
```

На компоненты приведенной выше схемы сообщения накладываются те же ограничения, что и на ответ на запрос Renew. Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

8.10 EnumerationEnd

Если источник данных неожиданно заканчивает перечисление, источнику данных следует отправить сообщение SOAP EnumerationEnd в ссылку оконечной точки, указанную при создании перечисления. Сообщение должно иметь следующую форму:

```
(1) <s:Envelope ...>
(2) <s:Header ...>
(3) <wsa:Action>
(4) gost.e-ecolog.ru
(5) </wsa:Action>
(6) <wsa:To>xs:anyURI</wsa:To>
(7)   ...
(8)  </s:Header>
(9)  <s:Body ...>
(10) <wsmen:EnumerationEnd ...>
(11) <wsmen:EnumerationContext>...</wsmen:EnumerationContext>
(12) <wsmen:Code>
(13)   [
(14) gost.e-ecolog.ru
(15) | gost.e-ecolog.ru
(16)   ]
(17) </wsmen:Code>
(18) <wsmen:Reason xml:lang="language identifier" >
(19)  xs:string
(20) </wsmen:Reason> ?
(21)   ...
(22) </wsmen:EnumerationEnd>
(23) </s:Body>
(24) </s:Envelope>
```

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Body/wsmen:Release/wsmen:EnumerationContext

Данный обязательный элемент содержит данные XML, которые представляют завершаемый контекст перечисления. Потребителям рекомендуется НЕ пытаться сравнивать данный элемент с любой коллекцией элементов wsmen:EnumerationContext в целях соотнесения, так как для этого требуется возможность сравнивать произвольные элементы XML. Если потребители желают найти корреляцию с их незавершенными контекстами, им рекомендуется использовать параметры ссылки/wsmen:Enumerate/wsmen:EndTo.

/s:Envelope/s:Body/wsmen:EnumerationEnd/wsmen:Code =

“gost.e-ecolog.ru

Эта значение должно использоваться, если источник данных закончил перечисление из-за закрытия контролируемым способом; то есть, если источник данных закрыт, но имеет возможность отправить сообщение EnumerationEnd, прежде чем отключится.

/s:Envelope/s:Body/wsmen:EnumerationEnd/wsmen:Code =

“gost.e-ecolog.ru

Это значение должно использоваться, если источник данных завершил перечисление по некоторой другой причине, прежде чем закончился срок его действия.

/s:Envelope/s:Body/wsmen:EnumerationEnd/wsmen:Reason

Данный необязательный элемент содержит текст на языке, определенном атрибутом @xml:lang, описывающий причину неожиданного завершения перечисления.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

9 Специализированные действия (методы)

Специализированные действия или «методы» являются обычными сообщениями SOAP с уникальными значениями Action. Реализация может поддерживать методы, специфичные для ресурса, в любой форме в рамках модели адресации и ограничений, описанных в разделе 5.

R9-1 Служба, соответствующая настоящему стандарту, может предоставить любые специализированные действия или методы.

R9-2 Если предоставлются специализированные методы, должны соблюдаться правила адресации, как описано в настоящем стандарте, и у каждого специализированного метода должен быть уникальный wsa:Action.

R9-3 Если запрос не содержит корректные параметры для специализированного действия, служба может вернуть сообщение-отказ wsman:InvalidParameter. Также могут быть включены коды детализации отказа для неправильного типа и неправильного имени.

gost.e-ecolog.ru (неправильный тип) http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/InvalidName (неправильное имя)

Как определено адресацией, URI Action используется для описания семантики операции, а элемент wsa:To описывает место назначения сообщения. Таким образом, специализированный метод имеет выделенный URI Action адресации.

Поскольку опции являются методом параметризации для типов сообщений, нерасширяемых для пользователей, такие как операции доступа к ресурсу, они не подходят для использования в качестве специализированного метода и не комбинируются с ним. Специализированные операции, определенные в документе WSDL, определяют любые требуемые параметры, и, таким образом, предоставляют строгое именование и проверку типа. Смешивание wsman:OptionSet со строго типизированной операцией WSDL, по всей вероятности, приведет к путанице.

10 Уведомления (обработка событий)

10.1 Общие положения

Часто инфраструктуры управления желают получать сообщения, когда в отдаленных службах и приложениях управления происходят события. Необходим механизм для регистрации интереса, так как набор Веб-служб, заинтересованных в получении таких сообщений, часто неизвестен заранее или изменяется с течением времени. Данная спецификация определяет ряд операций для одной Веб-службы управления (называемой «подписчик») для регистрации интереса (назваемого «подпиской») с другой Веб-службой управления (называемой «источником события») о получении сообщений о событиях (называемых «уведомлениями» или «сообщениями о событиях»). Подписчик может управлять подпиской, взаимодействуя с Веб-службой (называемой «менеджером подписки»), определяемой источником события.

Для повышения надежности источник события может ограничить подписчику срок действия подписки, при этом срок действия подписки может быть большим. Менеджер подписки обеспечивает возможность подписчику возобновить или отменить подписку, прежде чем ее срок действия закончится.

Есть много механизмов, с помощью которых источники события могут поставлять события приемникам событий. Данная спецификация обеспечивает расширяемый способ выбора предпочтительного механизма доставки подписчиками. В настоящем стандарте определен асинхронный механизм отправки подписчику, намерение заключается в том, что это не накладывает никаких ограничений на то, какие механизмы доставки могут поддерживаться данной спецификацией.

Чтобы создать, возобновить и удалить подписки, подписчики отправляют сообщения-запросы менеджерам подписки и источникам события.

Когда источник события принимает запрос о создании подписки, он обычно создает ее на определенное количество времени, хотя источник события может принять подписку без определенного срока действия. Если менеджер подписки принимает запрос о возобновлении, он обновляет срок действия. В течение этого времени источником событий поставляются уведомления для требуемого приемника событий. Источник события может поддерживать фильтрацию, чтобы ограничить уведомления, которые поставлены приемнику событий; в этом случае, а также при условии, что запрос содержит фильтр, источник события отправляет только уведомления, которые соответствуют требуемому фильтру. Источник события отправляет уведомления до тех пор, пока не произойдет одно из следующих событий: менеджер подписки принимает запрос на отказ от подписки, срок действия подписки истекает, так как она не возобновлялась, либо источник события досрочно отменяет подписку. В последнем случае источник события прилагает максимальные усилия к тому, чтобы указать причины окончания подписки.

При отсутствии надежной передачи сообщений на прикладном уровне (например, [WS-ReliableMessaging]) определенные здесь сообщения передаются с использованием качества обслуживания нижележащего транспортного протокола и без требований к резервированию и гарантиям доставки на прикладном уровне.

Если управляемая сущность генерирует события, возможно опубликование этих событий, используя парадигму и механизм издатель-подписчик (publish-and-subscribe).

R10.1-1 Если ресурс может генерировать события и позволяет клиентам подписываться и получать уведомления, он должен делать это посредством операций, определенных в данном разделе.

R10.1-2 Если поддерживается механизм обработки событий, описанный в данном разделе, должны поддерживаться сообщения wsme:Subscribe, wsme:Renew и wsme:Unsubscribe. Сообщение wsme:SubscriptionEnd является необязательным. Сообщение wsme:GetStatusне требуется от реализаций с ограниченными ресурсами. Если это сообщение не поддерживается, в ответ на запрос должен быть возвращен отказ wsa:ActionNotSupported.

10.2 Подписка (Subscribe)

В некоторых сценариях сам источник события управляет подписками, которые он создал. В других сценариях, например в географически-распределенной системе издатель-подписчик, полезным может оказаться делегирование управления подпиской другой Веб-службе. Чтобы поддерживать такую гибкость, ответ источника события на запрос о подписке включает EPR службы, с которой подписчик может взаимодействовать для управления этой подпиской. Будущие запросы о возобновлении или от-

мене подписки должны направляться на эту ERP. Ссылка может указывать на ту же самую Веб-службу (Address и ReferenceParameters), что и источник события, либо он может обращаться к некоторой другой Веб-службе, которой источник события делегировал управление данной подпиской; однако, полный EPR менеджера подписки (Address и ReferenceParameters) должен быть уникальным для каждой подписки.

В настоящем стандарте используется термин «менеджер подписки» для Веб-службы, управляющей подпиской, независимо от того, является ли она сама источником события или некоторой отдельной Веб-службой.

Чтобы создать подписку, подписчик отправляет источнику события сообщение-запрос следующей формы:

```
(1)  <s:Envelope ...>
(2)   <s:Header ...>
(3)    <wsa:Action>
(4)     gost.e-ecolog.ru
(5)    </wsa:Action>
(6)    ...
(7)   </s:Header>
(8)   <s:Body ...>
(9)    <wsme:Subscribe ...>
(10)     <wsme:EndTo>ссылка-на-оконечную-точку</wsme:EndTo> ?
(11)     <wsme:Delivery Mode="xs:anyURI"? >xs:any</wsme:Delivery>
(12)     <wsme:Expires>[xs:dateTime | xs:duration]</wsme:Expires> ?
(13)     <wsme:Filter Dialect="xs:anyURI"? > xs:any </wsme:Filter> ?
(14)     ...
(15)    </wsme:Subscribe>
(16)  </s:Body>
(17) </s:Envelope>
```

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Header/wsa:Action

Если в привязке SOAP используется URI действия SOAP, то для данного URI должно использоваться значение, указанное здесь.

/s:Envelope/s:Body / */wsme:EndTo

Куда отправить сообщение SubscriptionEnd, если срок подписки неожиданно истек. При наличии, данный элемент должен иметь тип wsa:EndpointReferenceType. По умолчанию данное сообщение не должно отправляться. Оконечная точка, на которую ссылается данный EPR, должна реализовать привязку «EndToEndpoint» вида portType, описанную в приложении И.

/s:Envelope/s:Body / */wsme:Delivery

Место назначения для уведомлений с использованием некоторого режима доставки.

/s:Envelope/s:Body / */wsme:Delivery/@Mode

Режим доставки, который будет использоваться для уведомлений, отправленных по этой подписке. Значение по умолчанию

«gost.e-ecolog.ru , которое указывает, что должна использоваться доставка в режиме без запросов.

Если источник события не поддерживает запрашиваемый режим доставки, то запрос должен быть отброшен, и источник события может отправить отказ wsme:DeliveryModeRequestedUnavailable, указывающий, что требуемый режим доставки не поддерживается.

/s:Envelope/s:Body/*/wsme:Delivery/@Mode=

«gost.e-ecolog.ru

Значением /s:Envelope/s:Body / */wsme:Delivery является единственный элемент NotifyTo, содержащий ссылку оконечной точки, в которую нужно отправить уведомления.

/s:Envelope/s:Body / */wsme:Expires

Требуемое время срока окончания для подписки. (Нет значения по умолчанию.) Источник события определяет фактический срок окончания и может использовать время меньшее или большее, чем запрошенный срок окончания. Время срока окончания может быть определенным временем или продолжительностью (длительность интервала времени с момента создания подписки). И явно заданное время, и время продолжительности интерпретируются по часам источника события.

Если данный элемент отсутствует, то запрос соответствует подписке без срока окончания. Таким образом, подписчик просит источник события создать подписку с неопределенным сроком жизни. Если источник события предоставляет такую подписку, она может быть завершена подписчиком посредством запроса Unsubscribe или источником события в любое время по причине завершения соединения, ограничения ресурсов или завершения работы системы.

Если срок окончания будет нулевой продолжительностью или конкретным временем, которое происходит в прошлом согласно источнику события, то запрос должен быть отброшен, и источник события может отправить отказ InvalidExpirationTime, указывающий на то, что было затребовано недействительное время срока окончания.

Некоторые источники события могут не иметь доступных «настенных часов» времени, и, таким образом, в состоянии принять срок окончания только в формате продолжительности. Если такой источник получит запрос Subscribe, содержащий явное время окончания, то запрос может быть не исполнен; в таком случае источник события может отправить отказ UnsupportedExpirationType, указывающий, что был запрошен неподдерживаемый тип срока окончания.

/s:Envelope/s:Body / */wsme:Filter

Булевское выражение, записанное на некотором диалекте как строка или как фрагмент XML. Если значение этого выражения равно false для некоторого уведомления, то это уведомление нельзя отправить приемнику событий. Предполагаемое значение — выражение, которое всегда равно true. Если источник событий не поддерживает фильтрацию, то запрос, содержащий фильтр, должен вызвать сообщение об ошибке, и источник события может отправить отказ wsme:FilteringNotSupported, указывающий, что фильтрация не поддерживается.

Если источник событий поддерживает фильтрацию, но не может выполнить требуемую фильтрацию, то запрос должен быть отброшен, и источник события может отправить отказ wsme:FilteringRequestedUnavailable, указывающий, что требуемый диалект фильтра не поддерживается.

/s:Envelope/s:Body/*/wsme:Filter/@Dialect

Значение по умолчаню «gost.e-ecolog.ru

Поскольку предикатные выражения XPath обеспечивают большую гибкость и силу, могут быть определены дополнительные диалекты фильтра. Например, более простой, менее сильный диалект может быть определен для реализаций с ограничением в ресурсах, или новый диалект может быть определен для поддержания фильтрации, основанной на данных, не включенных в само уведомление. При необходимости диалект фильтра может разрешить определение сложного фильтра, содержащего множество фильтров на других диалектах.

/s:Envelope/s:Body/*/wsme:Filter/@Dialect=«gost.e-ecolog.ru

Значение/s:Envelope/s:Body / */wsme:Filter является предикатным выражением XPath [XPath 1.0] (PredicateExpr); контекст выражения:

- узел контекста: Конверт SOAP, содержащий уведомление;
- положение в контексте: 1;
- размер контекста: 1;
- привязки переменных контекста: Нет;
- библиотеки функций: Основная библиотека функции [XPath 1.0];
- объявления пространства имен: свойство [in-scope namespaces][Инфо-набор XML] элемента /s:Envelope/s:Body / */wsme:Filter.

В запросы и ответы могут включаться другие информационные заголовки сообщения, определенные в адресации, согласно правилам использования и семантике, определенным в адресации.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если источник события принимает запрос о создании подписки, то он должен дать ответ следующей формы:

```
(1)  <s:Envelope ...>
(2)    <s:Header ...>
(3)      <wsa:Action>
(4)        gost.e-ecolog.ru
(5)      </wsa:Action>
(6)      ...
(7)    </s:Header>
(8)    <s:Body ...>
(9)      <wsme:SubscribeResponse ...>
(10)       <wsme:SubscriptionManager>
(11)         wsa:EndpointReferenceType
(12)       </wsme:SubscriptionManager>
(13)       <wsme:Expires>[xs:dateTime | xs:duration]</wsme:Expires>
(14)       ...
(15)     </wsme:SubscribeResponse>
(16)   </s:Body>
(17) </s:Envelope>
```

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/S:Header/wsa:RelatesTo

Значение совпадает со значением wsa:MessageID соответствующего запроса.

/s:Envelope/s:Body/*/wsme:SubscriptionManager

EPR менеджера подписки для данной подписки.

В некоторых случаях удобно, чтобы все EPR, выпущенные единственным источником события, адресовали единственную Веб-службу и использовали параметры ссылки для различения среди активных подписок. Для удобства в общей ситуации данная спецификация определяет глобальный элемент Identifier типа xs:anyURI, который может использоваться в качестве различающего параметра ссылки, если так пожелал источник события.

/s:Envelope/s:Body / */wsme:Expires

Срок окончания, установленный источником события

Время срока окончания может быть абсолютным временем или продолжительностью, того же типа, что и срок окончания в запросе (если таковой имеется).

Если данный элемент отсутствует, то у подписки не истекает срок окончания. Таким образом, у подписки есть неопределенный срок жизни. Она может быть завершена подписчиком посредством запроса Unsibscribe или источником события в любое время по причинам завершения соединения, ограничения ресурсов или завершения работы системы.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если источник события примет решение не принимать подписку, то запрос должен быть отброшен, и источник события может отправить отказ wsme:EventSourceUnableToProcess, указывающий, что запрос не был принят.

Данная спецификация не ограничивает уведомления, так как любое сообщение может быть уведомлением.

Однако, если приемник событий, запросивший подписку, хочет, чтобы уведомления были специально помечены, он может определить буквальные блоки заголовков SOAP в запросе Subscribe, в элементах/s:Envelope/s:Body/wsme:Subscribe/wsme:NotifyTo/wsa:ReferenceParameters; в соответствии с

правилами адресации, источник события должен включать каждый такой буквальный блок заголовка SOAP в каждое уведомление, отправленное в оконечную точку, на которую ссылается /s:Envelope/s:Body/wsme:Subscribe/wsme:NotifyTo.

10.2.1 Общие положения

WS-Management использует Subscribe преимущественно, как задокументировано здесь, за исключением того, что модель адресации WS-Management по умолчанию включена как описанно в 5.1.

R10.2.1-1 Идентичность источника события должна быть основана на адресации EPR.

R10.2.1-2 Если служба не может поддерживать требуемую адресацию, следует вернуть сообщение-отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Подтверждение применимости адреса позволяет обнаружить ошибки во время создания подписки. Например, если нельзя достичь адреса из-за конфигурации брандмауэра и служба может это обнаружить, сообщение клиенту позволяет немедленно исправить ошибку.

R10.2.1-3 Поскольку различные режимы доставки требуют установления отдельного соединения для доставки событий, если HTTP или HTTPS используются для доставки событий, службе следует соответствовать профилям безопасности, определенным в разделе 11. Если защита не определена, служба может попытаться использовать механизмы защиты по умолчанию или вернуть отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Так как клиентам может потребоваться, чтобы в каждую доставку событий включался клиентский контекст, с этой целью в блоке Delivery может использоваться адрес NotifyTo. ERP данного NotifyTo может содержать любое число определенных клиентами параметров ссылки.

R10.2.1-4 Служба может проверить адрес, попытавшись установить связь в процессе обработки запроса Subscribe, чтобы гарантировать успешную доставку. Если служба определяет, что адрес некорректен либо разрешения не могут быть получены, следует вернуть отказ wsman:EventDeliverToUnusable.

Такая ситуация может произойти, если указан неправильный адрес или в случае, когда источник события не может должным образом получить разрешение на доставку событий.

R10.2.1-5 Любые параметры ссылки, поставляемые в адресе NotifyTo, должны быть включены с каждой доставкой событий как заголовки верхнего уровня, как определено 5.4. Если EndTo поддерживается, к нему также применяется это положение.

При использовании службами модели адресации по умолчанию часто используется Resource URI для ссылки на логический источник события, и дополнительно могут использоваться значения селекторов, чтобы указать на реальную или виртуальную регистрацию в рамках этого источника или даже для ограничения типов или группы доступных событий. Это действие может логически наложиться на механизм Filter в самом теле подписки, поэтому необходимо уделить должное внимание взаимодействию адреса источника события, типам событий, которые он может публиковать, и фильтрации на уровне подписки.

Если клиенту требуется, чтобы события доставлялись более чем в одно место назначения, требуется более одной подписки.

R10.2.1-6 Если события содержат локализованное содержимое, службе следует принимать подписку с блоком wsman:Locale, действующим как указание (см. 6.3), в пределах блока Delivery сообщения Subscribe. Язык закодирован в атрибуте xml:lang — с использованием языковых кодов RFC 5646.

Служба старается локализовать любое описательное содержимое на указанном языке при доставке таких событий, что обозначается следующим образом:

```
(1)  <wsme:Subscribe>
(2)    <wsme:Delivery>
(3)      <wsme:NotifyTo>  ... </wsme:NotifyTo>
(4)      <wsman:Locale xml:lang="language-code"/>
(5)    </wsme:Delivery>
(6)  </wsme:Subscribe>
```

Примечание — В этом контексте элемент wsman:Locale (определенный в 6.3) не является заголовком SOAP, и mustUnderstand не может использоваться.

R10.2.1-7 Службе следует принимать подписку с блоком wsman:ContentEncoding в пределах блока Delivery сообщения Subscribe. Данный блок действует как указание на способ кодирования доставляемых событий. Для этой цели определены два стандартных символа xs:language: «UTF-8» и «UTF-16», хотя при необходимости могут быть определены другие форматы кодирования. Службе следует попытаться закодировать события, используя требуемый языковой символ, как в следующем примере:

Пример:

(1) <wsme:Subscribe>

(2) <wsme:Delivery>

(3) ...

(4) <wsme:NotifyTo> ... </wsme:NotifyTo>

(5) <wsman:ContentEncoding> UTF-16 </wsman:ContentEncoding>

(6) </wsme:Delivery>

(7) </wsme:Subscribe>

10.2.2 Фильтрация

Выражение фильтра должно быть булевским предикатом. Чтобы поддерживать произвольные специальные запросы, включая проекции, WS-Management определяет элемент wsman:Filter точно такой же формы, что и элемент, используемый в операции Subscribe, за исключением того, что выражение фильтра не обязано быть булевским предикатом. Это позволяет использовать подписки с использованием существующих языков запросов, таких как SQL и CQL, которые объединяют предикат и информацию о проекции в одном синтаксисе. Использование проектирования определено диалектом фильтра, а не спецификацией WS-Management.

Если диалектом фильтра либо для Filter, либо для wsman:Filter, используемого для сообщения Subscribe, является gost.e-ecolog.ru (диалект по умолчанию в обоих случаях), то узлом контекста будет элемент SOAP Envelope.

WS-Management определяет элемент wsman:Filter как дочерний элемент элемента Subscribe.

WS-Management определяет элемент wsman:Filter для разрешения проекций, который описывается следующим образом:

(1) <wsman:Filter Dialect="xs:anyURI"?> xs:any</wsman:Filter>

Атрибут Dialect является необязательным. Если он не указан, у него есть следующее значение по умолчаню:

gost.e-ecolog.ru R/1999/REC-xpath-19991116

Данный диалект позволяет использовать любое выражение полного XPath или подмножества.

R10.2.2-1 Если служба поддерживает подписки, использующие Filter, то она должна также поддерживать фильтрацию с использованием wsman:Filter. Это правило позволяет стекам клиента всегда выбирать пространство имен XML wsman для элемента Filter. Даже если служба поддерживает wsman:Filter, для поддержания проекции это не требуется.

R10.2.2-2 Если служба поддерживает подписки с фильтрацией, использующие wsman:Filter, она должна также поддерживать фильтрацию с использованием Filter.

R10.2.2-3 Если запрос Subscribe содержит и Filter, и wsman:Filter, служба должна вернуть сообщение-отказ wsa:InvalidMessage.

Для того чтобы было возможно определять выражения фильтра обработки событий независимо от режима доставки, WS-Management определяет новый диалект фильтра, который совпадает с ранее определенным, за исключением того, что узел контекста определен как элемент, который будет передан как первый дочерний элемент элемента Body SOAP, если используется режим доставки без запросов. URI этого диалекта фильтра:

gost.e-ecolog.ru

Узел контекста для этого выражения следующий:

- узел контекста: любой элемент XML, который может быть передан как прямой дочерний элемент элемента s:Body при режиме доставки без запросов;
- положение в контексте: 1;
- размер контекста: 1;
- привязки переменных: нет;
- библиотеки функций: Основная библиотека функции [XPath 1.0];

- объявления пространств имен: свойство [in-scope namespaces][Инфо-набор XML] элемента /s:Envelope/s:Body/wsme:Subscribe/wsman:Filter.

R10.2.2-4 Службам следует поддерживать данный диалект фильтра, когда они хотят использовать фильтр, основанный на XPath, а не диалект фильтра по умолчанию, определенный в 10.2.1.

Соображения, описанные в 8.3 относительно диалекта фильтра XPath 1.0, также относятся к указанному фильтру обработки событий.

Реализации, ограниченные по ресурсам, могут использовать подмножество синтаксиса XPath, несмотря на затруднения при обеспечении полной обработки XPath. Это не требует добавления нового диалекта, если выражение, определенное в фильтре, является истинным выражением XPath. Использование URI диалекта фильтра не подразумевает, что служба поддерживает всю спецификацию для этого диалекта, а только то, что выражение соответствует правилам этого диалекта. Большинство служб использует XPath только для фильтрации, и они не поддержат создание нового XML или удаление частей XML, что может привести к фрагменту XML, нарушающему XML-схему события.

Пример:

Типичный пример использования XPath в подписке. Предположим, что у каждого события, которое было доставлено, есть следующее содержимое XML:

(1) <s:Body>

(2) <LowDiskSpaceEvent xmlns="...">

(3) <LogicalDisk>C:</LogicalDisk>

(4) <CurrentMegabytes>12</CurrentMegabytes>

(5) <Megabytes24HoursAgo>17</Megabytes24HoursAgo>

(6) </LowDiskSpaceEvent>

(7) </s:Body>

Событие полностью содержится в рамках элемента s:Body сообщения SOAP. Точка привязки для оценки XPath — первый элемент каждого события, и оно не ссылается на элемент <на s:Body>, как таковой. Выражение XPath оценивается, как если бы содержимое события было отдельным XML-документом.

Пример:

При использовании для простой обработки документов следующие четыре выражения XPath «выбирают» весь узел <LowDiskSpaceEvent>:

(8) /

(9) /LowDiskSpaceEvent

(10) ../LowDiskSpaceEvent

(11) .

При использовании в качестве «фильтра» это выражение XPath не фильтрует экземпляры и совпадает с отбором всех экземпляров события или с полным отсутствием фильтра.

Пример:

Однако используя следующий синтаксис, выражение XPath выбирает узел XML только если проверяемое выражение в скобках оценивается как логически «верное»:

(1) ../LowDiskSpaceEvent [LogicalDisk = «C»:]

В этом случае событие отобрано, если оно относится к дисководу «С»; иначе узел XML не будет отобран. Это выражение XPath отфильтровывает все события <LowDiskSpaceEvent> для других дисков.

Пример:

Полные реализации XPath могут поддерживать более сложные проверяемые выражения:

(1) ../LowDiskSpaceEvent [LogicalDisk = "C": and CurrentMegabytes <"20"]

В сущности, XML-форма события логически пропускается через процессор XPath, чтобы проверить, должна ли она быть отобрана. Если да, то она представляется как событие. В противном случае от события отказываются и не поставляют подписчику.

XPath 1.0 может использоваться просто для фильтрации или для передачи подмножества представления (или даже значений без оберток XML). В случаях, где результат не просто отфильтрован, но и «изменен», применяется в 8.6.

Если полный XPath не может поддерживаться, с этой целью описано общее подмножество в приложении Г.

R10.2.2-5 Элемент wsman:Filter должен содержать простой текст или единственный элемент XML простого или составного типа. Службе следует отклонить любой фильтр со смешанным содержимым или несколькими равнозначными элементами XML, используя отказ wsme:EventSourceUnableToProcess.

R10.2.2-6 Служба, соответствующая настоящему стандарту, может не поддерживать весь синтаксис и вычислительную мощность указанного диалекта фильтра. Единственное требование — чтобы указанный фильтр был синтаксически корректным в рамках определения диалекта. Таким образом, подмножества являются правомерными. Если указанный фильтр превышает возможности службы, следует вернуть сообщение-отказ wsman:CannotProcessFilter с текстом, объясняющим, почему фильтр вызвал проблему.

R10.2.2-7 Если служба требует сложных параметров инициализации в дополнение к фильтру, они должны быть частью блока wsman:Filter, потому что логически являются частью инициализации фильтра, даже если некоторые параметры не используются в процессе фильтрации. В этом случае должен быть создан уникальный диалект URI для источника события, а также опубликованы схемы и сценарии использования.

R10.2.2-8 Если служба поддерживает композицию нового XML или фильтрацию до состояния,когда результирующее событие не соответствует исходной схеме для этого события, доставку событий следует оформлять таким же образом как и содержимое для операций доступа на уровне фрагмента (см. 7.7).

Независимо от того, как отфильтрованы или урезаны события, в момент фактической отправки они должны соответствовать некоторому определению схемы XML. Простая отсылка развернутых фрагментов XML во время доставки не допускается.

R10.2.2-9 Если служба требует определенный инициализационный XML в дополнение к фильтру для оформления подписки, то этот инициализационный XML должен быть частью тела фильтра и зарегистрирован как часть диалекта фильтра.

Это правило последовательно поддерживает местоположение для содержимого инициализации, которое может логически рассматриваться как часть фильтра. XML-схема фильтра более понятна, если она выделяет инициализацию и фильтрацию в отдельные элементы XML.

Для получения информации о фильтрации по перечислениям см. 8.3.

10.2.3 Повторные подключения

Из-за характера доставки событий подписчик может быть недоступен в момент возникновения события. Вместо того, чтобы немедленно завершить все подписки, служба, как правило, пытается соединиться несколько раз с подходящими интервалами.

R10.2.3-1 Служба может следовать любой политике повторных подключений или позволить подписчику определить ее включением в подписку следующего элемента wsman:ConnectionRetry. Если служба не принимает элемент wsman:ConnectionRetry, следует вернуть сообщение-отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Это относится только к неудачным попыткам соединиться и не включает повторное воспроизведение фактических доставок SOAP.

```
(1)  <wsme:Subscribe>
(2)    <wsme:Delivery>
(3)      <wsme:NotifyTo>  ... </wsme:NotifyTo>
(4)      <wsman:ConnectionRetry Total="count"> xs:duration
</wsman:ConnectionRetry>
(5)    </wsme:Delivery>
(6)  </wsme:Subscribe>
```

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsman:ConnectionRetry xs:duration — интервал между повторами при попытке соединения

wsman:ConnectionRetry /@Total — число попыток, с соблюдением указанного интервала между попытками

R10.2.3-2 Если число повторных попыток исчерпано, подписку следует считать некорректно завершенной.

Механизм повторных попыток применяется только к попыткам соединиться. Отказы в доставке на установленном соединении могут привести к завершению связи согласно правилам используемого транспортного протокола и завершению подписки. Другие механизмы Веб-служб могут использоваться для создания надежной доставки или безопасного повторного воспроизведения фактических доставок.

10.2.4 SubscribeResponse

Служба возвращает любые определенные службой параметры ссылки в SubscriptionManager EPR, и они включаются подписчиком (клиентом) позже в сообщениях Unsubscribe и Retry.

R10.2.4-1 В SubscribeResponse служба может определить любой EPR для SubscriptionManager. Однако рекомендуется, чтобы адрес содержал тот же самый адрес, что и wsa:To исходного запроса Subscribe, и отличался только другими частями адреса, такими как параметры ссылки.

R10.2.4-2 Служба, соответствующая настоящему стандарту, может не возвратить в ответе поле Expires, но, как определено в 10.2, это подразумевает, что подписка не истекает, пока явно не будет отменена.

10.2.5 Событие контроля времени (Heartbeat)

Типичной проблемой с подписками событий является ситуация, в которой не происходит передача событий. Клиентам трудно отличить отсутствие событий, соответствующих подписке от неудачи подписки, в результате которой они не смогли получить какие бы то ни было уведомления.

Поэтому WS-Management определяет псевдособытие «сердцебиение», которое можно периодически посылать для любой подписки. Это событие отправляют, когда не происходят никакие регулярные события, чтобы клиент знал, что его подписка все еще активна. Если событие «сердцебиение» не прибывает, клиент понимает, что соединение неисправно, или истекла подписка, и может принять меры для исправления ситуации.

Событие «сердцебиение» отправляют вместо события, которое произошло бы и никогда не смешивают с «реальными» событиями. Во всех режимах, включая пакетный, это событие происходит в одиночку.

Чтобы запросить событие «сердцебиение» в рамках подписке, запрос Subscribe имеет дополнительное поле в секции Delivery:

```
(1) <wsme:Delivery>
(2)   ...
(3)   <wsman:Heartbeats> xs:duration </wsman:Heartbeats>
(4)   ...
(5) </wsme:Delivery>
```

wsman:Heartbeats определяет, что события контроля времени добавлены к потоку событий с указанным интервалом.

R10.2.5-1 Службе следует поддерживать события контроля времени. Если служба не поддерживает их, то она должна вернуть сообщение-отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

События «сердцебиение» относятся ко всем режимам доставки.

События «сердцебиение» также применяется к доставкам в режиме «pull» (доставка по запросу), в этом они служат указанием издателю о том, как часто ожидать запрос Pull. Служба может отказаться поставлять события, если клиент не запрашивает регулярно с указанным интервалом. Если за указанный интервал не появились новые события, служба просто включает событие «сердцебиение» как результат Pull.

R10.2.5-2 В то время как подписка с контролем времени активна, служба должна гарантировать, что либо реальные события, либо «сердцебиение» отправлены в пределах указанного интервала wsman:Heartbeat. Служба может оправить сердцебиение в этом интервале в дополнение к другим событиям, в то время как события контроля времени отправляются отдельно (не скомпонованы с другими событиями). Цель состоит в том, чтобы гарантировать, что некоторое движение событий происходит всегда в рамках заданного интервала.

R10.2.5-3 Служба, соответствующая настоящему стандарту, может отсылать «сердцебиения» с меньшим интервалом, чем указано в подписке. Однако эти события не должны смешиваться с другими событиями, когда используются пакетные режимы доставки. Как правило, Heartbeat отправляется только тогда, когда не происходит никаких реальных событий. Если реальные события доставлялись в указанном интервале, служба может не производить в нем событие «сердцебиение».

R10.2.5-4 Служба, соответствующая настоящему стандарту, не должна отправлять событие «сердцебиение» одновременно с уже происходящими доставками событий. Они должны отправляться последовательно, как любые другие события, как единичные события или как единственное событие в пакете.

На практике события контроля времени основаны на таймере обратного отсчета. Если никакие события не имеют место, отправляется одно событие «сердцебиение». Однако всякий раз, когда отсылается реальное событие, перезагружается таймер обратного отсчета. Если происходит непрекращающийся поток событий, «сердцебиение» может вообще не поставляться.

«Сердцебиение» должно подтверждаться как любое другое событие, если используется режим доставки с подтверждением.

Клиент предполагает, что подписка больше не активна, если не получает события «сердцебиение» в пределах указанного интервала, таким образом, служба может отменить подписку и отправить любые запрашиваемые сообщения SubscriptionEnd, так как клиент, вероятно, вскоре подпишется снова. Если события контроля времени используются в сочетании с закладками (см. 10.2.6), то можно достигнуть очень надежной доставки с известным поведением задержек.

Само событие контроля времени — это просто сообщение о событии без тела, и это событие определяется его URI wsa:Action следующим образом:

```
(1)  <s:Envelope ...>
(2)    <s:Header>
(3)      <wsa:To> .... </wsa:To>
(4)      <wsa:Action s:mustUnderstand="true">
(5)        gost.e-ecolog.ru
(6)      </wsa:Action>
(7)      ...
(8)    </s:Header>
(9)    <s:Body/>
(10) </s:Envelope>
```

10.2.6 Закладки

Надежная доставка событий труднодостижима, поэтому у подписчиков управления должен быть метод, чтобы быть уверенными в получении всех событий из источника. Когда подписки истекают или происходит сбой доставки, может возникнуть временное окно, в котором клиент не будет уверен, произошло ли какое-то критическое событие. Вместо того, чтобы использовать очень сложную, транзактную модель доставки, WS-Management определяет простой механизм для обеспечения доставки всех событий или обнаружения пропущенных событий.

Данный механизм требует, чтобы источники события поддерживали краткосрочные или долгосрочные журналы событий. Клиент подписывается таким же образом, что и при обычной операции Subscribe, и определяет, что должны использоваться закладки. Тогда с каждой доставкой событий служба отправляет новую закладку, за хранение которой отвечает клиент. Эта закладка по существу является контекстом или указателем на логическое местоположение в потоке событий, которое соответствует фильтру подписки. Когда происходит каждая новая доставка, клиент обновляет закладку в своем собственном пространстве. Если подписка истекает или неожиданно закончилась, клиент может подписаться снова, используя последнюю известную закладку. В сущности, фильтр подписки определяет желаемый набор событий, а закладка сообщает службе откуда начинать запись. Тогда клиент может начать с того места, где ранее закончил.

Данный механизм невосприимчив к проблемам транзакций, поскольку клиент может просто начать с любой из нескольких недавних закладок. Единственное требование для службы — вести некоторый журнал событий, к которму применяется закладка. Если представленная закладка слишком

стара (по времени или позиции в пределах журнала), служба может отклонить запрос. Тогда клиент, по крайней мере, точно знает, что события были пропущены.

R10.2.6-1 Служба, соответствующая настоящему стандарту, может поддерживать механизм закладок WS-Management. Если служба не поддерживает закладки, следует вернуть сообщение-отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Чтобы запросить службы закладки, клиент включает элемент wsman:SendBookmarks в запрос Subscribe следующим образом:

```
(1)  <s:Body>
(2)    <wsme:Subscribe>
(3)      <wsme:Delivery>
(4)        ...
(5)      </wsme:Delivery>
(6)      <wsman:SendBookmarks/>
(7)    </wsme:Subscribe>
(8)  </s:Body>
```

wsman:SendBookmarks является инструцией службе отправить закладку с каждой доставкой событий. Закладки применяются во всех режимах доставки.

Закладка — это токен, который представляет абстрактный указатель в потоке событий, но при этом не имеет значения, указывает ли он на последнее поставленное событие, или последнее событие плюс один (предстоящее событие), потому что этот токен поставляется одной и той же реализации во время последующей операции Subscribe. Таким образом, служба может прикрепить к закладке любую семантику и структуру, определенные службой, без изменения на стороне клиента.

Если запрашиваются закладки, в каждой доставке событий содержится новое значение закладки как заголовок SOAP, как показано в следующей схеме. Формат закладки полностью определяется службой и рассматривается клиентом как непрозрачное значение.

```
(1)  <s:Envelope
(2)      xmlns:s="gost.e-ecolog.ru
(3)      xmlns:wsa="gost.e-ecolog.ru
(4)      xmlns:wsman="gost.e-ecolog.ru                                    >
(5)    <s:Header>
(6)      <wsa:To s:mustUnderstand="true">gost.e-ecolog.ru                 >
(7)      ...
(8)      <wsman:Bookmark> xs:any </wsman:Bookmark>
(9)      ...
(10)   </s:Header>
(11)   <s:Body>
(12)     ...содержимое события...
(13)   </s:Body>
(14) </s:Envelope>
```

wsman:Bookmark включает содержимое XML, поставляемое службой, которое указывает на логическую позицию данного события или пакета событий в потоке событий, подразумеваемом подпиской.

R10.2.6-2 Если закладки поддерживаются, то содержимое элемента wsman:Bookmark должно быть либо простым текстом, либо единственным составным элементом XML. Служба, соответствующая настоящему стандарту, не должна принимать смешанное содержимое из текста и элементов или нескольких дочерних элементов XML внутри элемента wsman:Bookmark.

R10.2.6-3 Если закладки поддерживаются, то служба должна использовать в заголовке элемент wsman:Bookmark для отправления обновленной закладки с каждой доставкой событий. Закладки сопровождают только доставку событий и не входят в сообщения SubscriptionEnd.

После того, как подписка закончилась по любой причине, последующее сообщение Subscribe со стороны клиента может включать закладку в запросе на подписку. Тогда служба узнает, где надо начинать.

Последняя известная закладка, полученная клиентом, добавляется к сообщению Subscribe как новый блок, размещенный после дочерних элементов Subscribe, как в следующей схеме:

```
(1)  <s:Body>
(2)    <wsme:Subscribe>
(3)      <wsme:Delivery>  ... </wsme:Delivery>
(4)      <wsme:Expires> ... </wsme:Expires>
(5)      <wsman:Filter> ... </wsman:Filter>
(6)      <wsman:Bookmark>
(7)        ...последняя известная закладка от предыдущей доставки...
(8)      </wsman:Bookmark>
(9)      <wsman:SendBookmarks/>
(10)   </wsme:Subscribe>
(11) </s:Body>
```

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsman:Bookmark — произвольное содержимое XML, ранее отправленное службой как wsman:Bookmark во время доставки событий от предыдущей подписки

wsman:SendBookmarks — инструкция продолжит доставку обновленных закладок с каждой доставкой событий

R10.2.6-4 Закладка — указатель на последнее доставленное событие или пакет событий. Служба должна возобновить доставку с первого события или событий после события, представленного закладкой. Служба не должна повторно воспроизводить события, связанные с закладкой, или пропускать события после закладки.

R10.2.6-5 Служба может поддерживать короткую очередь предыдущих закладок, позволив подписчику начать использование любой из нескольких предыдущих закладок. Если закладки поддерживаются, от службы требуется поддерживать только новую закладку, для которой доставка очевидно произошла.

R10.2.6-6 Если закладку нельзя обработать, служба должна выдать отказ wsman:InvalidBookmark с одним из кодов детализации:

- срок закладки истек (источник не может восстановиться и повторно воспроизвести от того момента):

gost.e-ecolog.ru

- формат неизвестен:

gost.e-ecolog.ru

Если созданы несколько новых подписок с использованием предыдущей закладки, служба может разрешить многократное повторное использование или может ограничить закладки единственным подписчиком и даже может ограничить время, в течение которого закладки могут использоваться, прежде чем стать недействительными.

Следующее, определенное заранее, зарезервированное значение закладки указывает, что подписка начинается в самой ранней точке в потоке событий, поддержанном издателем:

gost.e-ecolog.ru

Если подписка получена с данной закладкой, источник события воспроизводит все возможные события, соответствующие фильтру, и все события, которые впоследствии происходят для этого источника события. Отсутствие закладки означает «начать со следующего доступного события».

R10.2.6-7: Служба, соответствующая настоящему стандарту, может поддерживать зарезервированную закладку gost.e-ecolog.ru и не поддерживать любой другой тип закладки. Если поддерживается закладка gost.e-ecolog.ru bookmark/earliest, источнику событий следует отправить все предыдущие и последующие события, которые соответствуют фильтру, начиная с самого раннего такого события.

10.2.7 Режимы доставки

Несмотря на то, что общий шаблон асинхронных сообщений, основанных на событиях, чрезвычайно распространен, различные приложения часто требуют различных механизмов доставки сообщения о событии. Например, в некоторых случаях оптимальным будет простое асинхронное сообщение, а в других ситуациях лучше будет, если потребитель событий сможет запрашивать сообщения о событиях, чтобы управлять потоком и выбором времени прибытия сообщения. Некоторые потребители требуют, чтобы сообщения о событиях были обернуты в стандартный конверт SOAP «событие», в то время как другие предпочитают, чтобы сообщения были доставлены в развернутом виде. Некоторые потребители могут потребовать, чтобы сообщения о событиях были гарантированно доставлены, в то время как другие могут быть готовы принимать доставку событий без гарантий.

Чтобы поддерживать такой широкий спектр требований к доставке событий, данная спецификация вводит абстракцию, названную режимом доставки. Это понятие используется в качестве точки расширения так, чтобы источники события и потребители событий могли свободно создавать новые механизмы доставки, которые приспособлены к их специальным требованиям. Данная спецификация предоставляет минимальную поддержку для согласования режима доставки: источник события может предоставить список поддерживаемых режимов доставки в ответ на запрос о подписке, в котором запрошен неподдерживаемый режим доставки.

Реализация WS-Management может поддерживать множество режимов доставки событий.

По существу доставка состоит из следующих элементов:

- режим доставки (как упакованы события);
- адрес (транспорт и местоположение в сети);
- профиль аутентификации для использования при соединении или доставке события (защите).

Стандартные профили защиты описаны в разделе 12 и могут потребоваться для подписок, если служба нуждается в указаниях или других индикаторах, какую модель защиты использовать в момент отправки события.

Если режим доставки поддерживается, но фактически неприменим из-за конфигурации брандмауэра, служба может вернуть с этой целью сообщение-отказ wsme:DeliveryModeRequestedUnavailable с дополнительной детализацией.

R10.2.7-1 Для любого заданного транспортного протокола совместимой службе следует поддерживать, по крайней мере, один из следующих режимов доставки, чтобы взаимодействовать с типичными клиентами:

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

Режим доставки *не* подразумевает какой-либо определенный транспорт.

Режимы описывают поведение сообщения SOAP и не связаны с используемым транспортным протоколом. Режим доставки подразумевает определенный формат сообщения SOAP, таким образом, сообщение, которое не вписывается в этот формат, требует нового режима доставки.

R10.2.7-2 Адрес NotifyTo в сообщении Subscribe должен поддерживать единственный режим доставки.

Это требование для клиента, так как служба не может проверить, верно ли это утверждение. Если это требование не соблюдается клиентом, служба не может работать правильно. Если подписчик поддерживает много режимов доставки, адрес NotifyTo должен быть в некотором роде дифференцирован, например, путем добавления дополнительного параметра ссылки.

10.2.8 Action URI событий

Как правило, у каждого типа события есть свой собственный wsa:Action URI для быстрого определения и маршрузации события. Если тип события не определяет свой собственный wsa:Action URI, следующий URI может использоваться по умолчанию:

gost.e-ecolog.ru

Данный URI может использоваться в случаях, когда типы событий выведены в режиме реального времени из других источников и не опубликованы как события Веб-службы, и, таким образом, не имеют назначенного wsa:Action URI. Данная спецификация не устанавливает ограничения для wsa:Action URI для событий. Более определенный URI может использоваться для надежной маршрутизации. Во многих случаях фиксированная схема может использоваться для моделирования многих различных типов событий, в таком случае идентификатором события является просто поле в содержимом XML события.

URI в этом случае может указать на схему и не отличаться для всех различных идентификаторов события, которые могли бы произойти, или он может указать определенное событие путем добавления суффикса идентификатора события к wsa:Action URI. Данная спецификация не устанавливает ограничений для степени детализации URI, но внимательное рассмотрение указанных проблем является частью проектирования URI для событий.

10.2.9 Последовательность доставки и подтверждение

Режим доставки указывает, как служба будет обмениваться событиями с заинтересованными сторонами. Данный пункт подробно описывает режимы доставки.

10.2.9.1 Общие положения

Для некоторых типов событий важна упорядоченная и подтвержденная доставка, но для других типов событий порядок прибытия незначим. WS-Management определяет четыре стандартных режима доставки:

- gost.e-ecolog.ru

В этом режиме в каждом сообщении SOAP есть только одно событие, и нет никаких подтверждений или ответов SOAP. Служба может отправлять события подписчику асинхронно безотносительно любых событий, находящихся в пути. Данный режим полезен, если порядок событий не имеет значения, например для событий, содержащих текущий итог, в которых каждое новое событие может заменить предыдущее полностью, и для идентификации события достаточно метки времени.

- gost.e-ecolog.ru /wsman/PushWithAck

В этом режиме в каждом сообщении SOAP есть только одно событие, и следующее событие отправляется после получения подтверждения на предыдущее. Служба ставит в очередь все недоставленные события для подписчика и отправляет каждое новое событие только после подтверждения предыдущего.

- gost.e-ecolog.ru /wsman/Events

В этом режиме в каждом сообщении SOAP могут быть несколько событий, и каждый пакет событий подтверждается, прежде чем отправляется следующий. Служба ставит в очередь все события для подписки и поставляет их в этом порядке, поддерживая порядок в пакетах.

- gost.e-ecolog.ru /wsman/Pull

В этом режиме в каждом сообщени SOAP могут быть несколько событий, и каждый пакет событий подтверждается. Поскольку получатель использует Pull для синхронного получения события, подтверждение будет неявным. Поддерживается порядок доставки.

Упорядочение событий через подписки не подразумевается.

Модель подтверждения обсуждается в пункте 10.8.

10.2.9.2 Асинхронный режим доставки

При стандартном режиме доставки:

gost.e-ecolog.ru ,

каждая доставка состоит из единственного события. Никакое подтверждение не происходит, поэтому клиент не может отменить подписку, отправив сообщение об ошибке.

Таким образом подписки, созданные с этим режимом доставки, могут иметь короткую продолжительность, чтобы предотвратить ситуации, в которых доставку нельзя остановить, если информация SubscriptionManager из сообщения SubscribeResponse испорчена или потеряна.

Чтобы обеспечить быструю маршрутизацию событий, обязательный элемент wsa:Action URI в каждом сообщении о событии может отличаться для каждого типа события независимо от строгости типизации тела события.

R10.2.9.2-1 Служба может поддерживать режим доставки

gost.e-ecolog.ru

R10.2.9.2-2 Чтобы точно управлять обработкой слишком больших событий, служба может принимать следующие дополнительные инструкции в подписке:

```
(1)  <wsme:Delivery>
(2)    <wsme:NotifyTo> ... </wsme:NotifyTo>
(3)    ...
(4)    <wsman:MaxEnvelopeSize Policy="enumConstant">
(5)      xs:positiveInteger
(6)    </wsman:MaxEnvelopeSize>
```

(7) ...

(8) </wsme:Delivery>

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsme:Delivery/wsman:MaxEnvelopeSize

максимальное количество октетов для всего конверта SOAP в единственной доставке событий

wsme:Delivery/wsman:MaxEnvelopeSize/@Policy

необязательное значение с одной из следующих перечисленных величин:

- CancelSubscription: отменить на первом событии с превышенным размером;
- Skip: молча пропустить события с превышенным размером;
- Notify: уведомить подписчика, что события были пропущены, как определено в 10.9.

R10.2.9.2-3 Если запрашивается wsman:MaxEnvelopeSize, то служба не должна отправлять тело событий большего размера, чем указано в ограничении. По умолчанию отправляется уведомление подписчику, как определено в 10.9, если иначе не указано в подписке, и служба пытается продолжить доставку. Если событие превышает какие-либо внутренние максимумы по умолчанию, службе следует также попытаться уведомить, как определено в 10.9, но не заканчивать подписку, если иначе не определено в подписке. Если значение wsman:MaxEnvelopeSize будет для службы слишком большим, то следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

В отсутствие любых других инструкций Policy службы должны отправить подписчикам уведомления о пропущенных событиях, как определено в 10.9.

10.2.9.3 Асинхронный режим доставки с подтверждением

Данный режим доставки идентичен стандартному режиму «Push», за исключением того, что каждая доставка подтверждается. Каждая доставка содержит одно событие, и элемент wsa:Action указывает на тип события. Однако подтверждение, основанное на SOAP, происходит, как описано в 10.7.

URI режима доставки:

gost.e-ecolog.ru

В любом другом отношении, кроме URI режима доставки, данный режим идентичен режиму Push, как описано в 10.2.9.2.

R10.2.9.3-1 Службам следует поддерживать режим доставки gost.e-ecolog.ru
wsman/PushWithAck. Если режим доставки не поддерживается, следует вернуть сообщение-отказ wsme:DeliveryModeRequestedUnavailable.

10.2.9.4 Пакетный режим доставки

Группировка событий является эффективным способом минимизировать трафик событий от источников событий высокой интенсивности, не жертвуя своевременностью событий. WS-Management определяет пользовательский режим доставки событий, который позволяет источнику события собирать несколько исходящих сообщений о событиях в один конверт SOAP. Доставка всегда подтверждается с использованием модели, определенной в 10.7.

R10.2.9.4-1 Служба может поддерживать режим доставки

gost.e-ecolog.ru

Если этот режим доставки не поддерживается, следует вернуть сообщение-отказ wsme:DeliveryModeRequestedUnavailable.

Для этого режима доставки у элемента Delivery следующий формат:

(1)<wsme:Delivery Mode="htgost.e-ecolog.ru

(2) <wsme:NotifyTo>

(3) wsa:EndpointReferenceType

(4) </wsme:NotifyTo>

(5) <wsman:MaxElements> xs:positiveInteger </wsman:MaxElements> ?

(6) <wsman:MaxTime> xs:duration </wsman:MaxTime> ?

(7) <wsman:MaxEnvelopeSize Policy="enumConstant">

(8) xs:positiveInteger

(9) </wsman:MaxEnvelopeSize> ?

(10) </wsme:Delivery>

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsme:Delivery /@Mode

обязательный атрибут, который должен быть определен как

gost.e-ecolog.ru

wsme:Delivery/wsme:NotifyTo

обязательный элемент, который должен содержать EPR, куда нужно отправить сообщения о событиях для данной подписки

wsme:Delivery/wsman:MaxElements

необязательный элемент, содержащий положительное целое число, которое указывает на максимальное число событий, которые можно скомплектовать в единый конверт SOAP.

Ресурс не должен отправлять больше этого числа элементов в единственной доставке, хотя может отправить меньше.

wsme:Delivery/wsman:MaxEnvelopeSize — необязательный элемент, содержащий положительное целое число, которое указывает на максимальное количество октетов в конверте SOAP, используемом для доставки событий

wsman:MaxEnvelopeSize /@ Policy — необязательный атрибут с одной из следующих констант:

- CancelSubscription: отменить на первом событии с превышением размера
- Skip: молча пропустить события с превышением размера
- Notify: уведомить подписчика, о пропуске событий, как определено в 10.9.

wsme:Delivery/wsman:MaxTime — необязательный элемент, содержащий продолжительность, которая указывает на максимальный интервал времени, в течение которого служба может собирать события в пакет.

Значение wsman:MaxTime не может быть превышено интервалом между кодированием первого события в пакете и отправлением пакета для доставки. Некоторые реализации издателя могут выбрать более сложные схемы, в которых различные события, включенные в подписку, поставляются с различными задержками или различными приоритетами. В таких случаях для этих целей может быть разработан определенный диалект фильтра, который будет использоваться для описания инструкции к издателю. В таких случаях wsman:MaxTime может быть опущен, если это неприменимо; однако, если он присутствует, переопределяет все, что определено фильтром.

В отсутствие любых других инструкций в любой части подписки службы должны поставлять подписчикам уведомления о пропущенных событиях, как определено в 10.9.

Если клиент хочет обнаружить соответствующие значения для wsman:MaxElements или wsman:MaxEnvelopeSize, клиент может запросить метаданные, определенные для служб. Формат таких метаданных выходит за рамки настоящего стандарта.

R10.2.9.4-2 Если пакетный режим требуется в сообщении Subscribe и отсутствуют элементы MaxElements, MaxEnvelopeSize и MaxTime, служба может выбрать любые применимые значения по умолчанию. Применяются следующие сообщения об ошибке:

- если MaxElements не поддерживается, wsman:UnsupportedFeature возвращается со следующим кодом детализации отказа:

gost.e-ecolog.ru

- если MaxEnvelopeSize не поддерживается, wsman:UnsupportedFeature возвращается со следующим кодом детализации отказа:

gost.e-ecolog.ru

- если MaxTime не поддерживается, wsman:UnsupportedFeature возвращается со следующим кодом детализации отказа:

gost.e-ecolog.ru

- если MaxEnvelopeSize /@Policy не поддерживается, wsman:UnsupportedFeature возвращается со следующим кодом детализации отказа:

gost.e-ecolog.ru

R10.2.9.4-3 Если затребовано wsman:MaxEnvelopeSize, служба не должна отправлять тело событий размером свыше указанного предела. Поведение по умолчанию должно уведомить подписчика, как определено в 10.9, если в подписке на этот счет нет иных инструкций, и попытаться продолжить доставку. Если событие превышает какие-либо внутренние максимумы по умолчанию, службе следует также сделать попытку уведомления, как определено в 10.9, но не завершать подписку, если иначе не определено в подписке.

Если подписка была создана с использованием пакетного режима, все сообщения доставки событий должны иметь следующий формат:

```
(1)  <s:Envelope ...>
(2)    <s:Header>
(3)      ...
(4)      <wsa:Action>
(5)        gost.e-ecolog.ru
(6)      </wsa:Action>
(7)      ...
(8)    </s:Header>
(9)    <s:Body>
(10)     <wsman:Events>
(11)       <wsman:Event Action="Action URI события">
(12)         ...тело события...
(13)       </wsman:Event> +
(14)     </wsman:Events>
(15)   </s:Body>
(16) </s:Envelope>
```

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

s:Envelope/s:Header/wsa:Action

обязательный элемент, который должен быть определен как gost.e-ecolog.ru wsman/1/wsman/Events

s:Envelope/s:Body/wsman:Events/wsman:Event

обязательные элементы, которые должны содержать тело соответствующего сообщения о событии, как если бы wsman:Event было элементом s:Body

s:Envelope/s:Body/wsman:Events/wsman:Event/@Action

обязательный атрибут, который должен содержать wsa:Action URI, который использовался бы для содержащегося сообщения о событии.

R10.2.9.4-4 Если требуется пакетный режим, то доставка должна подтверждаться, как описано в 10.7.

Пропущенные события (как определено в 10.9) закодированы с остальными событиями.

Пример:

Следующий пример показывает параметры пакетирования, поставляемые для операции Subscribe. Служба получает инструкцию отправить не более 10 элементов в пакете, ждать не более 20 с с момента кодирования первого события, пока не будет отправлен весь пакет, и включить не более 8192 октетов в сообщение SOAP.

```
(1)  ...
(2)  <wsme:Delivery
(3)   Mode="gost.e-ecolog.ru                                    >
(4)   <wsme:NotifyTo>
(5)    <wsa:Address>gost.e-ecolog.ru                    >
(6)   </wsme:NotifyTo>
(7)   <wsman:MaxElements>10</wsman:MaxElements>
(8)   <wsman:MaxTime>PT20S</wsman:MaxTime>
(9)   <wsman:MaxEnvelopeSize>8192</wsman:MaxEnvelopeSize>
(10) </wsme:Delivery>
```

Пример:

Ниже приведен пример пакетной доставки, которая соответствует настоящему стандарту:

```
(1) <s:Envelope
(2)  xmlns:s="gost.e-ecolog.ru                    03/05/soap-envelope"
(3)  xmlns:wsa="gost.e-ecolog.ru
(4)  xmlns:wsman="gost.e-ecolog.ru
(5)  xmlns:wsme="gost.e-ecolog.ru                                    >
(6)  <s:Header>
(7)   <wsa:To s:mustUnderstand="true">gost.e-ecolog.ru               >
(8)   <wsa:Action>
(9)    http://schemas.dmtf.org/wbem/wsman/1/wsman/Events
(10)   </wsa:Action>
(11)   ...
(12)  </s:Header>
(13)  <s:Body>
(14)   <wsman:Events>
(15)    <wsman:Event
(16)     Action="gost.e-ecolog.ru                                    >
(17)    <DiskChange
(18)     xmlns="gost.e-ecolog.ru                                     >
(19)     <Drive> C: </Drive>
(20)     <FreeSpace> 802012911 </FreeSpace>
(21)    </DiskChange>
(22)    </wsman:Event>
(23)    <wsman:Event
(24)     Action="gost.e-ecolog.ru                                    >
(25)     <DiskChange
(26)     xmlns="gost.e-ecolog.ru                                     >
(27)     <Drive> D: </Drive>
(28)     <FreeSpace> 1402012913 </FreeSpace>
(29)     </DiskChange>
(30)    </wsman:Event>
(31)   </wsman:Events>
(32)  </s:Body>
(33) </s:Envelope>
```

URI действия в строке 9 определяет, что это пакет, который содержит различные события. Тела событий находятся в строках 15—22 и строках 23—30. Фактический атрибут Action для каждого события — атрибут обертки wsman:Event.

10.2.9.5 Режим доставки по запросу

В некоторых обстоятельствах запрос событий является эффективным способом управлять потоком данных и уравновешивать своевременность относительно возможности обработки. Кроме того, в некоторых случаях сетевые ограничения препятствуют использовать асинхронный режим Push; то есть служба не может установить соединение с подписчиком.

WS-Management определяет специальный режим доставки событий «по запросу», который позволяет источнику событий поддерживать логическую очередь сообщений о событиях, полученных перечислением. Данный режим доставки использует сообщение Pull, чтобы извлечь событие из логической очереди. Однако также могут использоваться все другие операции публикации-подписки, определенные в данной главе, (например, для того чтобы отменить подписку, используется Unsubscribe, а не Release).

Для этого режима доставки элемент Delivery имеет следующий формат:

(1) <wsme:Delivery Mode =“gost.e-ecolog.ru >

(2) ...

(3) </wsme:Delivery>

wsme:Delivery / Mode должен быть

gost.e-ecolog.ru

R10.2.9.5-1 Служба может поддерживать режим доставки

gost.e-ecolog.ru . Если будет затребован режим Pull, но он не поддерживается, то служба должна вернуть сообщение-отказ wsme:DeliveryModeRequestedUnavailable.

wsman:MaxElements, wsman:MaxEnvelopeSize, и wsman:MaxTime не применяются в сообщении Subscribe для данного режима доставки, потому что сообщение Pull содержит всю необходимую функциональность для управления группированием и временем ответов.

R10.2.9.5-2 Если подписка некорректно задает параметры, которые не совместимы с режимом доставки по запросу, службе следует вернуть отказ wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

R10.2.9.5-3 Если режим Pull будет затребован в сообщении Subscribe и источник события принимает запрос на подписку, то элемент SubscribeResponse в ответном сообщении должен содержать элемент EnumerationContext, подходящий для использования в последующей операции Pull.

Пример:

(1) <s:Body ...>

(2) <wsme:SubscribeResponse ...>

(3) <wsme:SubscriptionManager>

(4) wsa:EndpointReferenceType

(5) </wsme:SubscriptionManager>

(6) <wsme:Expires>[xs:dateTime | xs:duration]</wsme:Expires>

(7) <wsmen:EnumerationContext>...</wsmen:EnumerationContext>

(8) ...

(9) </wsme:SubscribeResponse>

(10) </s:Body>

Подписчик извлекает EnumerationContext и использует его после этого в запросах Pull.

R10.2.9.5-4 Если режим доставки по запросу будет активен, то сообщения Pull должны использовать EPR менеджера подписки, полученного из сообщения SubscribeResponse. Параметры ссылки EPR могут следовать модели адресации, специфичной для службы, но могут использовать модель адресации WS-Management по умолчанию, если это подходит.

R10.2.9.5-5 Если активен режим доставки по запросу и запрос Pull не возвращает события (потому что ни одно не произошло с момента последнего «Pull»), следует вернуть сообщение-отказ wsman:TimedOut. EnumerationContext все еще считается активным, и подписчик может продолжить отправлять запросы Pull с последним EnumerationContext, для которого фактически произошла доставка событий.

R10.2.9.5-6 Если активен режим доставки по запросу и на запрос Pull возвращены события, служба может возвратить обновленный EnumerationContext, как определено для Pull, и ожидается, что подписчик будет использовать обновления, если таковые имеются, в последующем Pull, как определено для операций Enumerate. Если закладки активны, они также могут быть возвращены в заголовке и также должны быть обновлены службой.

На практике, служба может фактически не менять EnumerationContext, но клиент не может зависеть от того, что контекст остается постоянным. Он обновляется если не фактически, то концептуально.

В режиме доставки по запросу запрос Pull управляет группированием. Если никакие значения по умолчанию не определены, размер пакета равен 1, а максимальный размер конверта и срок ожидания определены службой.

R10.2.9.5-7 Если активен режим доставки по запросу, то служба не должна возвращать элемент EndOfSequence в потоке событий, потому что в этом режиме не существует понятия «последнее событие». Вместо этого контекст перечисления должен стать недействительным, если подписка истекает или по какой-либо причине отменена.

R10.2.9.5-8 Если будет использоваться режим доставки по запросу, служба должна принимать wsman:MaxEnvelopeSize, используемый в Pull, в качестве ограничения на размер событий, которые могут быть отправлены.

Свойства группирования, используемые в пакетном режиме, не применяются к режиму Pull. Клиент управляет максимальным размером событий, используя обычные механизмы в режиме Pull.

10.3 Получение статуса подписки (GetStatus)

Для получения статуса подписки подписчик отправляет запрос следующей формы менеджеру подписки:

```
(1)  <s:Envelope ...>
(2)    <s:Header ...>
(3)      <wsa:Action>
(4)        gost.e-ecolog.ru
(5)      </wsa:Action>
(6)      ...
(7)    </s:Header>
(8)    <s:Body ...>
(9)      <wsme:GetStatus ...>
(10)       ...
(11)     </wsme:GetStatus>
(12)   </s:Body>
(13) </s:Envelope>
```

На компоненты приведенной выше схемы сообщения накладываются ограничения, аналогичные ограничениям на запрос о возобновлении подписки. Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если подписка будет действительна и срок окончания не истек, то менеджер подписки должен дать ответ следующей формы:

```
(1)  <s:Envelope ...>
(2)    <s:Header ...>
(3)      <wsa:Action>
(4)        gost.e-ecolog.ru
(5)      </wsa:Action>
(6)      ...
(7)    </s:Header>
(8)    <s:Body ...>
(9)      <wsme:GetStatusResponse ...>
(10)       <wsme:Expires>[xs:dateTime | xs:duration]</wsme:Expires> ?
(11)       ...
(12)     </wsme:GetStatusResponse>
(13)   </s:Body>
(14) </s:Envelope>
```

На компоненты приведенной выше схемы сообщения накладываются ограничения, аналогичные ограничениям на запрос о возобновлении подписки. Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Сообщение wsme:GetStatus является необязательным для WS-Management.

R10.3-1 wse:GetStatus может не поддерживатся реализациями с огранченными ресурсами. Если это сообщение не поддерживается, то в ответ на данный запрос должен возвратиться отказ wsa:ActionNotSupported.

Может быть реализована поддержка периодических «сердцебиений», а не сообщения wsme:GetStatus.

10.4 Отказ от подписки (Unsibscribe)

Несмотря на то, что срок окнчания подписки, в конце концов, истекает, для минимизации ресурсов приемнику событий, выступающему как подписчик, следует явным образом удалить подписку, если он больше не хочет получать уведомления, связанные с подпиской.

Чтобы явно удалить подписку, подписывающийся приемник событий отправляет запрос следующей формы менеджеру подписки:

```
(1)  <s:Envelope ...>
(2)    <s:Header ...>
(3)      <wsa:Action>
(4)        gost.e-ecolog.ru
(5)      </wsa:Action>
(6)      ...
(7)    </s:Header>
(8)    <s:Body>
(9)      <wsme:Unsubscribe ...>
(10)       ...
(11)     </wsme:Unsubscribe>
(12)   </s:Body>
(13) </s:Envelope>
```

На компоненты приведенной выше схемы сообщения накладываются ограничения аналогичные ограничениям на запрос о возобновлении подписки. Например, отказы, определенные для него, также определены для запроса об удалении подписки.

Если менеджер подписки примет запрос об удалении подписки, он должен дать ответ следующей формы:

```
(1) <s:Envelope ...>
(2)    <s:Header ...>
(3)      <wsa:Action>
(4)    gost.e-ecolog.ru
(5)      </wsa:Action>
(6)      <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
(7)      ...
(8)    </s:Header>
(9)    <s:Body />
(10) </s:Envelope>
```

Компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

R10.4-1 Если служба поддерживает Subscribe, она должна реализовать сообщения Unsubscribe и гарантировать, что доставка событий будет завершена, если сообщение будет принято как корректное. Доставка событий может происходить после ответа на сообщение Unsubscribe до момента остановки транспортных потоков.

R10.4-2 Служба может в одностороннем порядке отменить подписку по любой причине, включая внутренние таймеры, реконфигурацию или ненадежное соединение.

Клиенты должны быть готовы получить любые события, которые уже находятся в пути, даже после того, как они отправили сообщение Unsubscribe. У клиентов есть выбор либо отказаться от любых таких доставок, либо принимать их.

Для этого сообщения используется EPR, полученный в элементе SubscribeResponse из элемента SubscriptionManager.

10.5 Возобновление подписки (Renew)

Для обновления срока окончания для подписки, менеджеры подписки должны поддерживать запросы возобновления подписки.

Для возобновления подписки подписчик отправляет запрос следующей формы менеджеру подписки:

```
(1) <s:Envelope ...>
(2)   <s:Header ...>
(3)     <wsa:Action>
(4)       gost.e-ecolog.ru
(5)     </wsa:Action>
(6)     ...
(7)   </s:Header>
(8)   <s:Body ...>
(9)     <wsme:Renew ...>
(10)      <wsme:Expires>[xs:dateTime | xs:duration]</wsme:Expires> ?
(11)      ...
(12)    </wsme:Renew>
(13)  </s:Body>
(14) </s:Envelope>
```

На компоненты приведенной выше схемы сообщения накладываются ограничения, аналогичные ограничениям на запрос о создании подписки. Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Если менеджер подписки принимает просьбу возобновить подписку, то он должен дать ответ следующей формы:

```
(1)  <s:Envelope ...>
(2)    <s:Header ...>
(3)      <wsa:Action>
(4)        gost.e-ecolog.ru
(5)      </wsa:Action>
(6)      ...
(7)    </s:Header>
(8)    <s:Body ...>
(9)      <wsme:RenewResponse ...>
(10)       <wsme:Expires>[xs:dateTime | xs:duration]</wsme:Expires> ?
(11)       ...
(12)     </wsme:RenewResponse>
(13)   </s:Body>
(14) </s:Envelope>
```

На компоненты приведенной выше схемы сообщения накладываются ограничения, аналогичные ограничениям на запрос о создании подписки, со следующим (ими) дополнением (ями):

/s:Envelope/s:Body / */wsme:Expires

Если требуемое окончание срока — продолжительность, то подразумеваемое начало этой продолжительности — момент начала обработки менеджером запроса Renew.

Если менеджер подписки принимает решение не возобновлять данную подписку, он должен отклонить запрос, также менеджер подписки может отправить отказ wsme:UnableToRenew, указывающий на то, что возобновление не было принято.

Другие компоненты приведенной выше схемы сообщения не ограничены данной спецификацией.

Требуется обработка сообщения Renew, но не требуется гарантировать успех.

R10.5-1 Хотя служба, соответствующая настоящему стандарту, должна принимать сообщение Renew как корректное действие, она всегда может отменить запрос с отказом wsme:UnableToRenew, вынуждая клиента подписаться заново.

Renew не имеет никакого воздействия на текущие доставки, закладки, периодические сообщения «сердцебиения» или другую продолжающуюся деятельность. Оно просто расширяет срок жизни подписки.

EPR, используемый для этого сообщения, получены от элемента SubscribeResponse в элементе SubscriptionManager.

10.6 Окончание подписки SubscriptionEnd

Если источник событий неожиданно заканчивает подписку, источнику событий следует отправить сообщение SOAP SubscriptionEnd по ссылке оконечной точки, с которой подписка была создана. Сообщение должно иметь следующую форму:

```
(1) <s:Envelope ...>
(2)  <s:Header ...>
(3)   <wsa:Action>
(4)    gost.e-ecolog.ru
(5)   </wsa:Action> ?
(6)   ...
(7)  </s:Header>
(8)  <s:Body ...>
(9)   <wsme:SubscriptionEnd ...>
(10)    <wsme:SubscriptionManager>
(11)     ссылка-на-оконечную-точку
(12)    </wsme:SubscriptionManager>
(13)    <wsme:Status>
(14)     [
(15)    gost.e-ecolog.ru                                                    |
(16)    gost.e-ecolog.ru                                                        |
(17)    gost.e-ecolog.ru
(18)     ]
(19)    </wsme:Status>
(20)    <wsme:Reason xml:lang="language identifier" >xs:string</wsme:Reason> ?
(21)     ...
(22)    </wsme:SubscriptionEnd>
(23)    ...
(24)  </s:Body>
(25) </s:Envelope>
```

Ниже описаны дополнительные нормативные ограничения на приведенную выше схему сообщения:

/s:Envelope/s:Body/*/wsme:SubscriptionManager

Ссылка оконечной точки менеджера подписки

Приемникам событий рекомендовано игнорировать данный элемент, поскольку его использование требует возможности сравнивать EPR на равенства, в то время как такой механизм не существует. Приемникам событий рекомендовано использовать параметры ссылки в wsme:Subscribe/wsme:EndTo EPR, если они хотят установить соответствие этого сообщения со своими подписками.

/s:Envelope/s:Body/wsme:SubscriptionEnd/wsme:Status =
“gost.e-ecolog.ru

Это значение должно использоваться, если источник событий завершил подписку из-за проблем с доставкой уведомлений.

/s:Envelope/s:Body/wsme:SubscriptionEnd/wsme:Status =
“gost.e-ecolog.ru

Это значение должно использоваться, если источник событий завершил подписку, так как источник завершается управляемым образом (то есть, если источник событий завершает работу, но имеет возможность отправить сообщение SubscriptionEnd до момента отключения).

/s:Envelope/s:Body/wsme:SubscriptionEnd/wsme:Status =
“gost.e-ecolog.ru

Это значение должно использоваться, если источник событий завершил подписку по иной причине, прежде чем истек срок окончания подписки.

/s:Envelope/s:Body/wsme:SubscriptionEnd/wsme:Reason

Данный необязательный элемент содержит текст на языке, определенном атрибутом @xml:lang, описывающим причину неожиданного завершения подписки.

Другие информационные заголовки сообщения, определенные в 5.4, могут быть включены в сообщение согласно правилам использования и семантике, определенной в 5.4.

Другие компоненты приведенной выше схемы сообщенияне ограничены данной спецификацией.

Сообщение SubscriptionEnd необязательно для WS-Management. Фактически, это «последнее событие» для подписки. Поскольку основная цель сообщения состоит в том, чтобы предупредить подписчика, что подписка завершилась, оно не подходит для использования с доставкой в режиме по запросу.

R10.6-1 Служба, соответствующая настоящему стандарту, может реализовать сообщение SubscriptionEnd.

R10.6-2 Служба, соответствующая настоящему стандарту, не должна реализовать сообщение SubscriptionEnd, когда доставка событий осуществляется в режиме по запросу, как определено в 10.2.9.4.

R10.6-3 Если SubscriptionEnd поддерживается, то сообщение должно содержать любые параметры ссылки, определенные подписчиком в адресе EndTo в исходной подписке.

R10.6-4 Это правило преднамеренно оставлено незаполненным.

Если служба поставляет события по тому же самому соединению, что и операция Subscribe, клиент, как правило, знает, что подписка была завершена, так как самосоединение закрывается или завершается.

Когда соединение доставки отличается от соединения Subscribe, настоятельно рекомендуется использовать сообщение SubscriptionEnd; иначе клиент не имеет непосредственного способа знать, что подписка больше не активна.

10.7 Подтверждение доставки

Для гарантирования подтверждения доставки на уровне приложений подписчик может запросить, чтобы приемник событий физически подтверждал доставку событий вместо того, чтобы полагаться полностью на гарантии уровня транспортного протокола.

Другими словами, транспортный протокол, возможно, принял события для отправки, но в процессе не отправил их приемнику фактического события, а служба перейдет дальше к следующему набору событий. Системные отказы могут привести к пропуску событий. Поэтому необходим механизм для осуществления подтверждения на уровне сообщений. Это позволяет передавать подтверждения на уровень приложений, увеличивая надежность доставки событий.

Клиент выбирает доставку с подтверждением путем выбора такого режима доставки, в котором у каждого события есть ответ. В настоящем стандарте приведены два режима доставки с подтверждением:

- gost.e-ecolog.ru /wsman/PushWithAck
- gost.e-ecolog.ru /wsman/Events

R10.7-1 Служба, соответствующая настоящему стандарту, может поддерживать режимы доставки PushWithAck (без запроса с уведомлением) или Events (пакетный). Однако, если будет запрошен

любой из данных режимов доставки, для поддержания упорядоченной очереди событий служба должна ждать подтверждения от клиента, прежде чем отправить следующее событие, соответствующее подписке.

R10.7-2 Если для подписки выбран режим доставки с подтверждением, то служба должна включать следующие заголовки SOAP в каждую доставку событий:

(1) <s:Header>

(2) <wsa:ReplyTo>куда направлять подтверждение</wsa:ReplyTo>

(3) <wsman:AckRequested/>

(4) ...

(5) </s:Header>

Следующие определения обеспечивают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsa:ReplyTo,

адрес должен всегда присутствовать в доставке событий вследствие наличия wsman:AckRequested.

Клиент извлекает данный адрес и отправляет подтверждение в указанный EPR, как требуется адресацией.

wsman:AckRequested

без содержимого; требует, чтобы подписчик подтвердил все доставки, как описано в данном пункте ниже.

Клиент в таком случае отвечает на доставку подтверждением или отказом.

R10.7-3 Служба может запросить сообщение-подтверждение, используя блок wsman:AckRequested, и впоследствии ожидать сообщение gost.e-ecolog.ru . Если такое сообщение не получено в качестве ответа, служба может завершить подписку.

Формат сообщения подтверждения, возвращаемого приемником событий (получателем) источнику события, идентичен для всех режимов доставки. Как показано в следующей схеме, он содержит уникальный wsa:Action, и поле wsa:RelatesTo установлено равным MessageID в доставке событий, к которой оно применяется:

(1) <s:Envelope ...>

(2) <s:Header>

(3) ...

(4) <wsa:To> Ссылка на оконечную точку из поля ReplyTo события </wsa:To>

(5)<wsa:Action>httgost.e-ecolog.ru

(6) <wsa:RelatesTo>ID сообщения исходной доставки события </wsa:RelatesTo>

(7) ...

(8) </s:Header>

(9) <s:Body/>

(10) </s:Envelope>

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

s:Envelope/s:Header/wsa:Action

URI, который должен быть gost.e-ecolog.ru

s:Envelope/s:Header/wsa:RelatesTo

элемент, который должен содержать wsa:MessageID доставки событий, к которой он относится. wsa:RelatesTo, является критическим элементом, который гарантирует, что подтверждается правильная доставка, и, таким образом, он не должен быть опущен.

s:Envelope/s:Header/wsa:To

Адрес EPR, извлеченный из поля ReplyTo в доставке событий

Все параметры ссылки должны быть также извлечены и добавлены к заголовку SOAP.

Несмотря на запрос о подтверждении, приемник событий может отказаться от доставки с отказом или может не ответить подтверждением. В этом случае источник событий может завершить подписку и отправить любые применимые сообщения SubscriptionEnd.

Если приемник событий не поддерживает подтверждение, он может ответить отказом wsman:UnsupportedFeature со следующим кодом детализации:

gost.e-ecolog.ru

Однако это действие столь же сложное, что и подтверждение доставки, поэтому большинство клиентов могут искать поле wsman:AckRequested и быть готовы подтвердить доставку или отказаться от нее.

10.8 Отказ от доставки

В любом из режимов доставки с подтверждениями, описанных в 10.7, приемник событий может отказаться получать доставку событий из соображений безопасности или по причине изменения политики. Тогда он отвечает отказом, а не подтверждением.

В этом случае источнику событий необходимо быть готовым завершить подписку даже при том, что подписчик не отправил сообщение Unsubscribe.

R10.8-1 Во время доставки событий, если приемник возвращает в ответ надоставку отказ wsman:DeliveryRefused, служба должна немедленно отменить подписку и может также отправить сообщение SubscriptionEnd к оконечной точке EndTo в оригинальной подписке, если такая функция поддерживается.

Таким образом, получатель может использовать отказы для отмены подписки, когда у него нет информации о SubscriptionManager.

10.9 Пропущенные события

События, которые не могут быть доставлены, не должны быть незаметно для подписчика исключены из потока событий, так как в таком случае подписчик может получить ложную картину истории событий. WS-Management определяет три функциональных возможности для событий, которые не могут быть доставлены в режиме push или которые слишком велики и не соответствуют ограничениям доставки, которые требует подписчик:

- завершить подписку;
- пропустить такие события;
- отправить специальное событие вместо пропущенных событий.

Эти варианты описаны в 10.2.9.2 и 10.2.9.3.

Во время доставки службе, возможно, придется пропустить события по следующим причинам:

- события превышают максимальный размер, который требует подписчик;
- клиент не успевает за потоком событий, и есть отставание;
- служба, возможно, была переконфигурирована или перезапущена, и события необратимо утеряны.

В этих случаях служба может сообщить клиенту, что события были пропущены.

R10.9-1 Если служба пропускает события, следует сгенерировать событие gost.e-ecolog.ru wbem/wsman/1/wsman/DroppedEvents, которое указывает клиенту, что часть событий пропущена. Любые параметры ссылки, определенные в адресе NotifyTo в подписке, должны также быть скопированы в это сообщение. Это событие — нормальная и неявно подразумеваемая часть любой подписки.

R10.9-2 Если отправлено событие gost.e-ecolog.ru , оно должно занять порядковую позицию исходного пропущенного события в потоке доставки. Событие DroppedEvents рассматривают как любое другое событие относительно его местоположения и других функциональных возможностей (закладки, аутентифицированная доставка, местоположение в пакете, и т. д.). Оно просто занимает место события, которое было пропущено.

Пример:

(1) <s:Envelope ...>

(2) <s:Header>

(3) ... ссылка-на-оконечную-точку подписчика...

(4)

(5) <wsa:Action>

(6) gost.e-ecolog.ru

```
(7)    </wsa:Action>
(8)   </s:Header>
(9)   <s:Body>
(10)  <wsman:DroppedEvents Action="URI wsa:Action пропущенного события">
(11)     xs:int
(12)    </wsman:DroppedEvents>
(13)    ...
(14)   </s:Body>
(15)  </s:Envelope>
```

Следующие определения обеспечивают дополнительные нормативные ограничения на приведенную выше схему сообщения:

s:Envelope/s:Header/wsa:Action — URI, который должен быть определен как gost.e-ecolog.ru org/wbem/wsman/1/wsman/DroppedEvents

s:Body/wsman:DroppedEvents / wsa:Action

ActionURI события, которое было пропущено.

s:Body/wsman:DroppedEvents — положительное целое число, которое представляет общее количество пропущенных событий начиная с момента создания подписки.

Операция Renew не имеет никакого эффекта на текущее общее количество пропущенных событий. Пропущенные события походят на любые другие события и могут потребовать подтверждения, повлиять на местоположение закладки и т. д.

Пример:

Ниже приведен пример пропущенного события в пакетной доставки событий:

```
(1)  <wsman:Events>
(2)   <wsman:Event Action="gost.e-ecolog.ru                    >
(3)    ...event body
(4)   </wsman:Event>
(5)   <wsman:Event
(6)    Action="gost.e-ecolog.ru                                          >
(7)   <wsman:DroppedEvents Action="gost.e-ecolog.ru                    ">
(8)    1
(9)   </wsman:DroppedEvents>
(10)  </wsman:Event>
(11)  <wsman:Event Action="gost.e-ecolog.ru                    >
(12)   ...тело события...
(13)  </wsman:Event>
(14) <wsman:Events>
```

R10.9-3 Если служба не может доставить событие и не поддерживает событие gost.e-ecolog.ru org/wbem/wsman/1/wsman/DroppedEvents, ей следует прекратить подписку вместо того, чтобы пропускать события без уведомления подписчика.

Поскольку это требование не может быть строго обязательным и некоторые пропущенные события не существенны, так как могут быть заменены последующим событием (например, текущий общий итог), требование сигнализировать о пропущенных событиях или прекращать подписку не являются строгими.

10.10 Управление доступом

Для источников события важно должным образом авторизовать запросы. Это особенно верно для запросов Subscribe, потому что возможность подписаться от имени стороннего получателя событий может использоваться для создания распределенной атаки «отказ в обслуживании».

Некоторые возможные схемы валидации запросов Subscribe включают:

- отправку получателю событий сообщение с описанием требуемой подписки, с последующим ожиданием подтверждающего сообщения, которое будет возвращено получателем событий, прежде

чем источник событий примет запрос подписки. В то время как данный метод обеспечивает сильную гарантию, что получатель событий фактически желает требуемую подписку, он не работает для получателей событий, которые не поддерживают отправку таких подтверждений, и требует дополнительную логику на стороне получателя событий.

- требование аутентификации пользователя для запроса Subscribe и позволения подписываться только авторизованным пользователям.

Возможны и другие механизмы.

Примечание — У источников события, которые не достижимы из Интернета, меньше потребность управлять запросами Subscribe.

10.11 Рекомендации по реализации

Реализациям следует генерировать срок окончания для запросов Subscribe и Renew и ответов на них, значительно превосходящий ожидаемую сетевую задержку.

Получателям событий следует быть готовыми получить уведомления о событиях после отправки запроса Subscribe, но до получения ответа на сообщения Subscribe. Получателям событий также следует быть готовыми получить уведомления после получения ответного сообщения Unsubscribe.

10.12 Анонсирование уведомлений

Источник события может выбрать анонсировать, какие виды уведомлений он может отправлять, включив хорошо определенный тип portType «EventSink» в свой WSDL. Подписчики могут исследовать указанный portType, чтобы определить, какие сообщения им, возможно, нужно поддерживать. Каждое уведомление появляется как независимая операция в пределах portType, как показано в следующем примере:

Пример:

```
(1)  <wsdl:portType name="EventSink">
(2)    <wsdl:operation name="WeatherReport">
(3)      <wsdl:input message="wr:ThunderStormMessage"
(4)        wsa:Action="urn:weatherReport:ThunderStorm"
(5)        wsam:Action="urn:weatherReport:ThunderStorm" />
(6)     <wsdl:input message="wr:TyphoonMessage"
(7)       wsa:Action="urn:weatherReport:Typhoon"
(8)       wsam:Action="urn:weatherReport:Typhoon" />
(9)    </wsdl:operation>
(10)  </wsdl:portType>>
```

В предыдущем примере данный источник событий может отправить два типа уведомлений (сообщения ThunderStorm и Typhoon).

Если не указано иное, получателям событий следует предполагать, что уведомления будут высылаться, используя SOAP 1.2 и с буквальным кодированием документа.

11 Метаданные и обнаружение

Протокол WS-management совместим со многими методами обнаружения ресурсов, доступных через службу.

Кроме того, настоящий стандарт определяет простую операцию вида запрос-ответ, облегчающую процесс установления связи с реализацией службы WS-Management в сетевой среде без предшествующего знания версии протокола, поддержанной реализацией. Эта операция используется для обнаружения службы, совместимой с WS-Management, в предположении, что известен транспортный адрес, по которому может быть передано сообщение. Как правило, может использоваться простой адрес HTTP.

Для гарантирования прямой совместимости настоящий стандарт определяет содержимое сообщения в пространстве имен XML, отдельном от пространства имен ядра протокола, которое не изменится по мере развития протокола. Далее, эта операция не зависит ни от какого заголовка конверта SOAP или содержимого тела кроме типов, явно определенных для данной операции. Таким образом, клиенты WS-Management могут быть уверены в возможности использовать эту операцию для всех ре-

ализаций и версий для того, чтобы убедиться в присутствии службы WS-Management, не зная заранее версии или характеристик поддерживаемого протокола.

Сообщение-запрос определено следующим образом:

```
(1)  <s:Envelope
(2)    xmlns:s="gost.e-ecolog.ru
(3)    xmlns:wsmid="gost.e-ecolog.ru                                    >
(4)    <s:Header>
(5)      ...
(6)    </s:Header>
(7)    <s:Body>
(8)      <wsmid:Identify>
(9)        ...
(10)     </wsmid:Identify>
(11)   </s:Body>
(12) </s:Envelope>
```

Следующие определения обеспечивают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsmid:Identify — тело запроса операции Identify, которое может содержать дополнительное содержимое, определенное поставщиком реализации, но, как правило, пусто.

Присутствие этого элемента тела составляет запрос.

Примечание — Отсутствуют пространства имен Addressing, пространства имен WS-Management и других понятий, зависящих от версии. Это сообщение совместимо только с основной спецификацией SOAP, и присутствие блока wsmid:Identify в s:Body есть воплощение операции в запросе.

Ответное сообщение определено следующим образом:

```
(13) <s:Envelope
(14) xmlns:s="gost.e-ecolog.ru
(15) xmlns:wsmid="gost.e-ecolog.ru                                    >
(16) <s:Header>
(17)  ...
(18) </s:Header>
(19) <s:Body>
(20) <wsmid:IdentifyResponse>
(21) <wsmid:ProtocolVersion> xs:anyURI </wsmid:ProtocolVersion> +
(22) <wsmid:ProductVendor> xs:string </wsmid:ProductVendor> ?
(23) <wsmid:ProductVersion> xs:string </wsmid:ProductVersion> ?
(24) <wsmid:InitiativeSupport>
(25) <wsmid:InitiativeName> xs:string </wsmid:InitiativeName> ?
(26) <wsmid:InitiativeVersion> xs:string </wsmid:InitiativeVersion> ?
(27) </wsmid:InitiativeSupport> ?
(28) <wsmid:SecurityProfiles>
(29) <wsmid:SecurityProfileName> xs:anyURI </wsmid:SecurityProfileName> *
(30) </wsmid:SecurityProfiles> ?
(31) <wsmid:AddressingVersionURI> xs:anyURI </wsmid:AddressingVersionURI> *
(32)  ...
(33) </wsmid:IdentifyResponse>
(34) </s:Body>
(35) </s:Envelope>
```

Следующие определения обеспечивают дополнительные, нормативные ограничения на приведенную выше схему сообщения:

wsmid:IdentifyResponse — тело ответа, содержащего метаданные реализации WS-Management;

wsmid:IdentifyResponse/wsmid:ProtocolVersion — обязательный элемент или элементы, в каждом из которых содержит URI, значение которого должно быть равно основному пространству имен XML, которое определяет поддерживаемую версию спецификации WS-Management;

Для каждой поддерживаемой версии протокола должен быть представлен один элемент. Службам следует также включать URI пространства имен XML для поддержанных зависимых спецификаций, таких как спецификация адресации. Например, если будущая версия WS-Management поддерживает несколько версий адресации, элемент IdentifyResponse может указать, какая из версий поддерживается.

wsmid:IdentifyResponse/wsmid:ProductVendor — необязательный элемент, который определяет поставщика реализации службы WS-Management посредством признанного имени или символа, такого как официальное название компании поставщика или его биржевой символ.

Также могут использоваться имя DNS, адрес электронной почты или URL Интернета.

wsmid:IdentifyResponse/wsmid:ProductVersion — необязательная строка версии реализации WS-Management.

Данная спецификация не налагает ограничения на формат или содержимое этого элемента.

wsmid:IdentifyResponse/wsmid:InitiativeSupport — необязательный элемент, который определяет инициативу, поддерживаемую реализацией WS-Management;

wsmid:IdentifyResponse/wsmid:InitiativeSupport/wsmid:InitiativeName — элемент, который определяет название инициативы, поддерживаемой реализацией WS-Management;

wsmid:IdentifyResponse/wsmid:InitiativeSupport/wsmid:InitiativeVersion — элемент, который определяет версию инициативы, поддерживаемой реализацией WS-Management.

Кроме того, за стандартизированными элементами может следовать содержимое, определенное поставщиком. После такого содержимого могут следовать следующие элементы:

wsmid:IdentifyResponse/wsmid:SecurityProfiles — необязательный элемент, который определяет набор профилей безопасности, поддерживаемых реализацией WS-Management;

wsmid:IdentifyResponse/wsmid:SecurityProfiles/wsmid:SecurityProfileName — необязательный элемент; его содержимое является URI, который определяет профиль безопасности, поддержанный реализацией WS-Management;

wsmid:IdentifyResponse/wsmid:AddressingVersionURI — необязательный элемент, представляет собой URI, который определяет версию Addressing, поддерживаемую реализацией WS-Management.

При поддержке службой данного элемента значение должно быть URI пространства имен XML-схемы, используемой версии адресации. Пространства имен XML-схемы, используемые в настоящем стандарте, перечислены в Приложении А. Служба может поддерживать и анонсировать несколько версий адресации.

R11-1 Службам WS-Management следует поддерживать операцию wsmid:Identify. Реализация службы, которая поддерживает данную операцию, должна сделать так независимо от версий WS-Management, поддержанных службой. Операция должна быть доступной по тем же адресам транспортного уровня, по которым доступны ресурсы.

Рекомендуется, чтобы клиентские приложения не включали какие-либо заголовки SOAP в операцию wsmid:Identify, доставленную по транспортному адресу, для которого делается запрос. Если присутствуют элементы заголовка SOAP, атрибут s:mustUnderstand на всех таких элементах может быть установлен в «false». В противном случае уменьшается вероятность успешного, независимого от версии ответа от службы.

R11-2 Служба, которая поддерживает операцию wsmid:Identify, не должна требовать присутствия каких-либо элементов заголовка SOAP для выполнения запроса. Если служба получает операцию wsmid:Identify, которая содержит неожиданное или неподдерживаемое содержимое заголовка с атрибутом s:mustUnderstand, равным «false», служба не должна отклонять запрос и должна обработать тело запроса как будто элементы заголовка отсутствуют.

R11-3 Служба, которая обрабатывает запрос wsmid:Identify, не должна требовать наличие каких-либо значений заголовков адресации, включая URI wsa:Action.

Вся цель этого механизма состоит в том, чтобы быть в состоянии определить присутствие определенных версий WS-Management (и соответствующих зависимых протоколов) независимым от версии способом.

Поскольку адресация не используется, адрес, к которому передано это сообщение, определен полностью на транспортном уровне и не представлен в содержании SOAP.

Если у клиента нет предварительных знаний о службе, включая идентификационные данные, желательно позволить службе обрабатывать сообщения Identify, не требуя аутентификации.

R11-4 Служба, которая поддерживает операцию wsmid:Identify, может предоставить эту операцию, не требуя аутентификации клиента или аутентификации сервера для обработки сообщения. В отсутствие других требований рекомендуется, чтобы сетевой адрес был дополнен суфиксом /wsman-anon/identify.

Службы, которые поддерживают неаутентифицированные запросы wsmid:Identify, могут выбрать не предоставлять описание протокола, поставщика или иную информацию о версии, которая может потенциально представлять уязвимость или способствовать уязвимости. Чтобы поддержать этот сценарий, данная спецификация определяет URI, службы которой могут использовать вместо URI фактической версии протокола WS-Management. Это значение может быть возвращено как значение элемента wsmid:ProtocolVersion сообщения wsmid:IdentifyResponse.

R11-5 Служба, поддерживающая неаутентифицированные сообщения wsmid:Identify, может вернуть следующий URI в качестве значения элемента wsmid:ProtocolVersion:gost.e-ecolog.ru
wsman/identity/1/wsmanidentity/NoAnonymousDisclosure.

R11-6 Служба, которая обеспечивает неаутентифицированный доступ к операции wsmid:Identify, но которая не сообщает в ответ на такие запросы фактические версии протокола WS-Management, поддерживаемые службой, должна поддерживать аутентифицированный доступ к операции wsmid:Identify. Такие службы должны возвращать в ответ на аутентифицированные запросы идентификаторы версии протокола WS-Management для каждой версии протокола WS-Management, поддерживаемой службой.

12 Безопасность

12.1 Общие положения

В целом операции управления и ответы должны быть защищены от атак, таких как неавторизованный доступ, перехват, повторное вопроизведение и изменение во время передачи. Аутентификация пользователя, отправившего запрос, также как правило необходима, чтобы можно было применить правила управления доступом для принятия решения об обработке запроса.

Настоящий стандарт устанавливает минимальные стандарты интероперабельности и предопределенные профили, использующие защиту транспортного уровня.

Данный подход обеспечивает наилучший баланс между простыми реализациями (стеки HTTP и HTTPS легко доступны, даже для аппаратных средств) и механизмами защиты, которые расположены перед любой обработкой сообщения SOAP, ограничивая возможности атаки.

Ожидается, что будут определены и использоваться более современный транспорт и профили уровня SOAP, изданные отдельно от настоящего стандарта.

Реализации, которые ожидают, что могут взаимодействовать друг с другом, могут адаптировать один или больше транспортов и моделей обеспечения безопасности, определенных в этом пункте, и могут определить любые дополнительные профили под различными обозначениями, основанными на URI.

12.2 Профили безопасности

Для настоящего стандарта профиль — произвольная комбинация транспорта или поведения SOAP, которая имеет общие потребности безопасности. В некоторых случаях профиль определен для целей документации и метаданных, но не мог бы быть частью фактического обмена сообщениями. Скорее это *описывает* вовлеченный обмен сообщениями.

Для обнаружения профилей, поддерживаемых службой могут использоваться метаданные, но это выходит за рамки данной конкретной спецификации.

Для всех предопределенных профилей за целостность сообщений, защиту, идентификацию и безопасность ответственен транспорт.

Профили аутентификации не появляются в трафике SOAP, за исключением сообщений Subscribe, использующих режимы доставки, которые требуют создания нового соединения источника события до получателя событий (например, режимы без запроса и пакетный). При создании подписки подписчик должен указать метод аутентификации для доставки, так как получатель событий должен подтвердить подлинность источника события (действующего как издатель) во время доставки событий.

В настоящем стандарте профили безопасности используют URI в качестве идентификатора. По мере определения профилей им можно присваивать URI и публиковать. WS-Management определяет ряд стандартизированных профилей безопасности для распространенных транспортных средств HTTP и HTTPS, как описано в Приложении В.3.1.

12.3 Соображения безопасности для подписок событий

Когда в подписках есть определение NotifyTo, часто важно указать службе, какую модель аутентификации использовать при доставке событий.

Если указания не присутствуют, служба может просто вывести из адреса wsa:To, что необходимо сделать. Однако, если служба может поддерживать различные режимы и имеет хранилище сертификатов или паролей, она может не знать, какую модель аутентификации выбрать или какие идентификационные данные использовать без указания в подписке.

WS-Management обеспечивает механизм по умолчанию для передачи желаемого режима аутентификации и идентификационных данных. Однако более сложные механизмы выходят за рамки данной версии WS-Management. Например, служба получателя событий может экспортировать метаданные, которые описывают доступные параметры, позволяя издателю согласовать соответствующую опцию. Дополнительные профили могут определить другие механизмы со своими заголовками SOAP с mustUnderstand =«true». WS-Management определяет дополнительное поле в блоке Delivery, которое может передать информацию для аутентификации, как показано в следующей схеме:

1) <s:Body>
(2) <wsme:Subscribe>
(3) <wsme:Delivery>
(4) <wsme:NotifyTo> Delivery EPR </wsme:NotifyTo>
(5) <wsman:Auth Profile="URI профиля аутентификации"/>
(6) </wsme:Delivery>
(7) </wsme:Subscribe>
(8) </s:Body>

Следующие определения накладывают дополнительные нормативные ограничения на приведенную выше схему сообщения:

wsman:Auth — блок с информацией для аутентификации, которая будет использоваться службами (действующими как издатель) для подтверждения подлинности получателя событий во время доставки событий;

wsman:Auth/@Profile — URI, который указывает, какой профиль безопасности использовать при установлении соединения для доставки событий.

Если блок wsman:Auth не присутствует по умолчанию, служба определяет необходимые действия при помощи адреса NotifyTo и любой предварительно сконфигурированной политики или параметров настройки, имеющихся у нее. Если блок wsman:Auth присутствует и никакие связанные с безопасностью токены не сообщены, служба должна знать,какие идентификационные данные использовать из собственной внутренней конфигурации.

Если служба уже настроена использовать определенный сертификат для доставки события, подписчик может запросить стандартную взаимную аутентификацию, как показано в следующей схеме:

(1) <s:Body>
(2) <wsme:Subscribe>
(3) <wsme:Delivery>
(4) <wsme:NotifyTo> HTTPS address </wsme:NotifyTo>
(5) <wsman:Auth
(6) Profile="gost.e-ecolog.ru >
(7) </wsme:Delivery>
(8) </wsme:Subscribe>
(9) </s:Body>

Если служба знает, как получить надлежащее имя пользователя и пароль для доставки событий, могут использоваться простые базовая или дайджест-аутентификация HTTP, как показано в следующей схеме:

(1) <s:Body>
(2) <wsme:Subscribe>
(3) <wsme:Delivery>
(4) <wsme:NotifyTo>адрес HTTP </wsme:NotifyTo>
(5) <wsman:Auth
(6) Profile="gost.e-ecolog.ru >
(7) </wsme:Delivery>
(8) </wsme:Subscribe>
(9) </s:Body>

Службы не должны поддерживать какой-либо определенный профиль. Далее в данном пункте определяются специализированные профили для доставки событий, в которых для службы нужна дополнительная информация, чтобы выбрать надлежащие идентификационные данные для использования при доставке событий.

12.4 Использование идентификационных данных при подписке

Данный пункт преднамеренно оставлен незаполненным.

12.5 Соотнесение событий с подпиской

Во многих случаях подписчик захочет гарантировать, что доставка событий соответствует действительной подписке и выпущена авторизованной стороной. В этом случае рекомендуется, чтобы в определение NotifyTo были введены параметры ссылки (ReferenceParameter).

Пример:

Во время подписки UUID может поставляться как идентификатор подписки:

(1) <s:Body>
(2) <wsme:Subscribe>
(3) <wsme:Delivery>
(4) <wsme:NotifyTo>
(5) <wsa:Address> address <wsa:Address>
(6) <wsa:ReferenceParameters>
(7) <MyNamespace:uuid>
(8) uuid:b0f685ec-e5c9-41b5-b91c-7f580419093e
(9) </MyNamespace:uuid>
(10) </wsa:ReferenceParameters>
(11) </wsme:NotifyTo>
(12) ...
(13) </wsme:Delivery>
(14) ...
(15) </wsme:Subscribe>
(16) </s:Body>

Это определение требует, чтобы служба включала значение MyNamespace:uuid как заголовок SOAP с каждым событием (см. пункт 5.1). Служба может использовать это значение, чтобы соотнести событие с подпиской, которой событие соответствует, и подтвердить происхождение события.

Это является механизмом аутентификации не транспортного уровня или уровня SOAP как таковой, но действительно помогает вести и синхронизировать списки подписок и определить, разрешена ли доставка событий даже при том, что само соединение, возможно, было аутентифицировано.

Данный механизм, тем не менее, может потребовать присутствие блока wsman:Auth для определения механизма защиты для аутентификации соединения на момент отправки события.

Каждая новая подписка может получить по крайней мере один уникальный одноразовый параметр ссылки, значимый для механизма, такой как UUID в примере.

Могут присутствовать другие параметры ссылки для маршрутизации доставки событий и соотнесения событий в соответствии с требованиями подписчика.

12.6 Ошибка аутентификации транспортного уровня

Поскольку транспортные средства, как правило, задействуют свои собственные механизмы аутентификации до передачи сообщений SOAP, первая попытка установления соединения может привести к ошибке аутентификации транспортного уровня. В таких случаях отказы SOAP не возникнут, и средство доставки отказа клиенту является реализационно- и транспортно-зависимым.

12.7 Соображения о защищенности сторонних подписок

Без надлежащей аутентификации и авторизации реализации WS-Management могут быть уязвимы для распределенных атак вида «отказ в обслуживании» через сторонние подписки на события. Эта уязвимость описана в 10.10.

13 Транспортные средства и кодирование сообщения

Данный пункт описывает правила кодирования, которые относятся ко всем транспортным средствам.

13.1 SOAP

WS-Management определяет использование SOAP, как определено в данном пункте.

R13.1-1 Служба должна, по крайней мере, получать и отправлять конверты SOAP в соответствии с SOAP 1.2.

R13.1-2 Служба может отклонить конверт SOAP, чей размер превышает 32,767 октетов.

R13.1-3 Службе не следует посылать конверт SOAP, чей размер превышает 32,767 октетов, если клиент не определил заголовок wsman:MaxEnvelopeSize, переопределяющий данный предел.

Ожидается, что большие конверты SOAP будут сериализоваться с использованием вложений.

R13.1-4 Любое сообщение-запрос может быть закодировано с использованием кодировок Unicode 3.0 (UTF-16) или UTF-8. Служба должна принимать кодировку UTF-8 для всех операций, также следует принимать UTF-16.

R13.1-5 Служба должна порождать ответы, используя ту же самую кодировку, что и исходный запрос. Если служба не поддерживает требуемую кодировку или не может определить кодировку, следует использовать кодировку UTF-8 в сообщении-отказе wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

R13.1-6 Для кодировки UTF-8 служба может не обработать сообщение, начинающееся с BOM UTF-8 (0xEF 0xBB 0xBF) и должна отправить ответы UTF-8 без BOM.

Присутствие BOM в 8-битных кодировках символов создает риск для функциональной совместимости. В случаях, когда требуются расширенные символы, может использоваться UTF-16.

R13.1-7 Если используется кодировка UTF-16, то служба должна поддерживать отметку порядка байтов (BOM) U+FFEB (обратный порядок байтов) или U+FFFE (прямой порядок байтов), как определено в спецификации Unicode 3.0, в качестве первого символа в сообщении (см. часто задаваемые вопросы Unicode BOM).

R13.1-8 Если информация о кодировании в BOM противоречит заголовку набора символов HTTP (charset) или если информация не полностью определяет кодировку, то служба должна вернуть отказ HTTP «некорректный запрос» (400).

Повторные заголовки с тем же самым QName, но различными значениями, которые подразумевают противоречащие функциональные возможности, считаются ошибкой клиентской стороны обмена сообщениями. Возвращение отказа помогает опознать клиентов с дефектами. Однако реализация может иметь ограниченные ресурсы и быть неспособной обнаружить повторные заголовки, таким образом, повторные заголовки могут быть проигнорированы. Повторные заголовки с тем же самым QName,

которые содержат информационные или непротиворечивые инструкции, возможны, но ни один такой заголовок не определен данной спецификацией, от которой она зависит.

R13.1-9 Если запрос содержит несколько заголовков SOAP с одинаковым QName из WS-Management, адресации или раздела 10, службе не следует обрабатывать их и следует вернуть отказ wsa:InvalidMessageInformationHeaders, если они обнаружены. (разделы 7 и 8 не определяют заголовки SOAP).

R13.1-10 По умолчанию службе, соответствующей настоящему стандарту, не следует отклонять запросы с начальными и завершающими пробелами в значениях элементов XML и по умолчаню следует удалять такие пробелы как будто их не было. Службам не следует порождать сообщения, содержащие начальные или завершающие пробелы в пределах значений элемента, если значение пробела не является частью значения. Если служба не может принимать использование пробела в рамках сообщения по причине того, что XML-схема устанавливает другое использование пробела, службе следует отправить отказ wsman:EncodingLimit со следующим кодом детализации: gost.e-ecolog.ru
wsman/1/wsman/faultDetail/Whitespace.

Клиенты могут отправлять сообщения с начальными или завершающими пробелами в значениях, и службы могут удалять «косметические» пробелы с обеих сторон значения элемента без отказа (см. Часть 2 XML-схемы: типы данных).

R13.1-11 Службам не следует отклонять сообщения, содержащие XML-комментарии, так как это — часть стандарта XML. Службы могут порождать сообщения, которые содержат комментарии, касающиеся происхождения и обработки сообщения, или добавляют комментарии для целей отладки.

13.2 Отсутствие ответа

Если операция завершается успешно, но ответ не может быть вычислен или фактически доставлен из-за трудностей во время выполнения или транспортных проблем, ответ не посылают и завершают соединение.

Такое поведение более предпочтительно, чем сложные модели для отправки ответов отсроченным способом. Реализации могут обычно хранить журнал всех запросов и их результатов и позволить клиенту повторно соединиться позже, чтобы получить содержимое журнала (используя Enumerate), если он не получил ответ. Формат и функциональные возможности такого журнала выходят за рамки настоящего стандарта. В любом случае клиент должен быть запрограммирован так, чтобы принять во внимание отсутствие ответа; все условия аномальных сообщений сводимы к этому сценарию.

R13.2-1 Если правильные ответы или отказы не могут быть вычислены или отправлены из-за внутренней ошибки службы, ответ не следует посылать.

Независимо, клиент должен обрабатывать случаи отсутствия ответа, таким образом, служба может просто вынудить клиента перейти в этот режим, а не отправлять ответ или отказ, который не определен в настоящем стандарте.

13.3 Повторное воспроизведение сообщений

Данный пункт преднамеренно оставлен незаполненным.

R13.3-1 Это правило преднамеренно оставлено незаполненным.

13.4 Пределы кодирования

Большинство следующих пределов определены в символах. Однако максимальный полный размер конверта SOAP определен в октетах. Реализациям разрешено превысить эти пределы. Служба считается соответствующей настоящему стандарту, если она соблюдает указанные пределы. Любое нарушение предела приводит к отказу wsman:EncodingLimit.

R13.4-1 Служба может не обрабатывать URI длиной, превышающей 2048 символов, и службе следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:
gost.e-ecolog.ru URILimitExceeded

R13.4-2 Службе не следует порождать URI длиной, превышающей 2048 символов.

R13.4-3 Служба может не обрабатывать имя опции длиной, превышающей 2048 символов.

R13.4-4 Служба может не обрабатывать значение опции длиной, превышающей 4096 символов.

R13.4-5 Служба может отклонить любую операцию, которая требует единственного ответа, превышающего 32,767 октетов.

R13.4-6 Служба может всегда отправлять отказы размером не более 4096 октет, вне зависимости от требований клиента ограничить размер ответа. Клиенты должны быть готовы к этому минимуму в случае ошибки.

R13.4-7 Когда используется модель адресации по умолчанию, служба может не обработать имя селектора, в котором больше, чем 2048 символов.

R13.4-8 У службы может быть максимальное количество селекторов, которые она может обработать. Если запрос содержит больше селекторов, чем данный предел, следует вернуть отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

R13.4-9 У службы может быть максимальное количество опций, которые она может обработать. Если запрос содержит больше опций, чем данный предел, следует вернуть отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

13.5 Двоичные вложения

Механизм оптимизации передачи SOAP (SOAP Message Transmission Optimization Mechanism, MTOM) используется для поддержки двоичных вложений в WS-Management. Если служба поддерживает вложения, применяются следующие правила:

R13.5-1 Служба, соответствующая настоящему стандарту, может произвольно поддерживать двоичные вложения в любой операции, используя предложение SOAP MTOM.

R13.5-2 Если служба поддерживает вложения, то служба должна поддерживать абстрактную функцию оптимизации передачи.

R13.5-3 Если служба поддерживает вложения, то служба должна поддерживать функцию оптимизированной сериализации MIME.

Другие типы вложений не запрещены. Определенные транспортные средства могут наложить дополнительные правила кодирования.

13.6 Чувствительность к регистру

В то время как XML и SOAP изначально чувствительны к регистру элементов схемы, WS-Management может использоваться со многими системами, которые не являются чувствительными к регистру. Эта поддержка прежде всего относится к значениям, но может также относиться к схемам, которые автоматически и динамически сгенерированы из других источников.

Служба может реализовать любое использование регистра, требуемое нижележащей средой выполнения.

Единственное требование заключается в том, что сообщения в состоянии пройти тесты валидации для определений схемы. В любое время средство валидации может быть установлено между клиентом и сервером в форме прокси, таким образом требование соответствия схеме является практическим.

В остальных случаях данная спецификация не налагает требования относительно использования регистра. Служба может интерпретировать значения как чувствительным к регистру, так и нечувствительным к регистру способом.

Рекомендуется, чтобы использование регистра не менялось при передаче какой-либо частью цепи обработки WS-Management. Использование регистра, установленное отправителем сообщения, должно сохраняться на протяжении всего времени жизни сообщения.

14 Отказы

Многие операции, определенные в WS-Managemente, могут генерировать отказы. Данный пункт описывает, как эти отказы должны быть преобразованы в сообщения SOAP.

14.1 Введение

Отказы возникают, когда сообщение SOAP успешно доставлено транспортом и обработано службой, но сообщение не может быть обработано должным образом. Если транспорт не может доставить сообщение к процессору SOAP, то происходит ошибка транспорта.

R14.1-1 Службе следует поддерживать только отказы SOAP 1.2 (или более новой спецификации).

В общем случае отказы не должны отправляться, если они не ожидаются как часть шаблона «Запрос — ответ». Например, клиент не должен посылать отказ на сообщение GetResponse, полученный в ответ на запрос Get.

14.2 Кодирование отказов

Данный пункт обсуждает кодирование отказа средствами XML.

R14.2-1 Служба, соответствующая настоящему стандарту, должна использовать следующий формат кодировки отказа и нормативные ограничения для отказов в пространстве WS-Management или любой из его зависимых спецификаций:

```
(1)  <s:Envelope>
(2)   xmlns:s="gost.e-ecolog.ru
(3)   xmlns:wsa="gost.e-ecolog.ru                                    >
(4)   <s:Header>
(5)   <wsa:Action>
(6)   gost.e-ecolog.ru
(7)   <wsa:Action>
(8)  <wsa:MessageID>
(9)  uuid:d9726315-bc91-430b-9ed8-ce5ffb858a87
(10)  </wsa:MessageID>
(11)  <wsa:RelatesTo>
(12)   uuid:d9726315-bc91-430b-9ed8-ce5ffb858a85
(13)   </wsa:RelatesTo>
(14)   </s:Header>
(15)
(16)   <s:Body>
(17)  <s:Fault>
(18)   <s:Code>
(19)   <s:Value> [Код] </s:Value>
(20)   <s:Subcode>
(21)   <s:Value> [Подкод] </s:Value>
(22)   </s:Subcode>
(23)   </s:Code>
(24)   <s:Reason>
(25)   <s:Text xml:lang="ru">  [Причина] </s:Text>
(26)   </s:Reason>
(27)   <s:Detail>
(28)   [Подробные сведения]
(29)   </s:Detail>
(30)   </s:Fault>
(31)   </s:Body>
(32)  </s:Envelope>
```

Следующие определения обеспечивают дополнительные нормативные ограничения на приведенную выше схему сообщения:

s:Envelope/s:Header/wsa:Action — корректное ActionURI отказа из соответствующей спецификации, в которой определен отказ;

s:Envelope/s:Header/wsa:MessageId — элемент должен присутствовать в сообщении-отказе, как и в любом сообщении без отказа;

s:Envelope/s:Header/wsa:RelatesTo — элемент, который, как любой другой ответ, должен содержать MessageID исходного запроса, вызвавшего отказ;

s:Body/s:Fault/s:Value — элемент должен быть или s:Sender или s:Receiver, как определено в пункте 14.6 для поля «Код»;

s:Body/s:Fault/s:Subcode/s:Value — для сообщений WS-Management равно одному из QNames подкодов, определенных в 14.6.

Если служба предоставляет специализированные методы или другие сообщения, это значение может быть другим QName, не из списка основных отказов, описанных в пункте 14.6.

s:Body/s:Fault/s:Reason — необязательный элемент, который содержит локализованный текст, объясняющий отказ более подробно.

Как правило, данный текст извлечен из поля «Причины» из соответствующей таблицы основных отказов (см. 14.6). Однако текст может быть адаптировать, чтобы отразить определенное обстоятельство. Данный элемент может быть повторен для различных языков. В элементе s:Text должен присутствовать атрибут xml:lang.

s:Body/s:Fault/s:Detail — необязательный элемент, отражающий рекомендуемое содержимое из соответствующей таблицы основных отказов (см. 14.6).

Шаблон отказа, приведенный выше, заполняется на основании записей в таблицах основных отказов из 14.6, который включает все соответствующие отказы от WS-Management и его нижележащих спецификаций.

s:Reason и s:Detail всегда небязательны, но рекомендуется их включать. Кроме того, s:Reason/s:Text содержит атрибут xml:lang, чтобы указать на язык, используемый в описательном тексте.

R14.2-2 Значение wsa:Action URI зависит от отказа. Служба должна отправить отказ, используя правильный URI, основываясь на спецификации, которая определила отказ. У отказов, определенных в настоящем стандарте, должно быть следующее значение URI: gost.e-ecolog.ru wsman/fault.

Таблицы основных отказов в 14.6 содержат соответствующий wsa:Action URI. Значение URI непосредственно подразумеваются именем QName отказа.

14.3 Отказы NotUnderstood

Есть особый случай отказов, касающихся атрибута mustUnderstand в заголовках SOAP. Спецификация SOAP определяет отказ иначе, чем кодирование в пункте 14.2 (см. подпункт 5.4.8 в SOAP 1.2). На практике отказ варьируется только в указании на заголовок SOAP, который не был понят, QName и пространство имен (см. строку 5 в следующей схеме).

```
(1)  <s:Envelope xmlns:s="gost.e-ecolog.ru
(2)   xmlns:wsa="gost.e-ecolog.ru                                    >
(3)
(4)   <s:Header>
(5) <s:NotUnderstood qname="имя QName заголовка" xmlns:ns="пространство имен XML заголовка"/>
(6)   <wsa:Action>
(7)    gost.e-ecolog.ru
(8)   </wsa:Action>
(9)   <wsa:MessageID>
(10)   urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a87
(11)  </wsa:MessageID>
(12)  <wsa:RelatesTo>
(13)   urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a85
(14)  </wsa:RelatesTo>
(15)  </s:Header>
(16)
(17)  <s:Body>
(18)  <s:Fault>
```

```
(19) <s:Code>
(20) <s:Value>s:MustUnderstand</s:Value>
(21) </s:Code>
(22) <s:Reason>
(23) <s:Text xml:lang="ru-RU">Заголовок не понят</s:Text>
(24) </s:Reason>
(25) </s:Fault>
(26) </s:Body>
(27)
(28) </s:Envelope>
```

Приведенный шаблон отказа может использоваться во всех случаях ошибки при обработке атрибута mustUnderstand. Строки 5—8 содержат важное содержимое, которое указывает, какой заголовок не был понят, и включает универсальный wsa:Action, который определяет, что данное сообщение — отказ.

Элемент wsa:RelatesTo включен, чтобы клиент мог сопоставить отказ с исходным запросом. Для транспортных средств,отличных от HTTP, в которых могут быть перепутаны запросы, это может быть единственным способом ответить правильному отправителю.

Если сам wsa:MessageID исходного запроса содержит ошибку и соединение использует протокол, ориентированный на запросы-ответы, служба может попытаться передать отказ обратно без поля wsa:RelatesTo или просто может не ответить, как описано в 14.4.

14.4 Вырожденные отказы

В редких случаях сообщение SOAP может не содержать достаточно информации, чтобы должным образом отправить отказ. Например, если wsa:MessageID искажен, то служба испытает затруднения с указанием ссылки на исходное сообщение при возвращении отказа. Некоторые транспортные средства могут быть не в состоянии сослаться на отправителя, чтобы возвратить отказ.

Если транспорт гарантирует простой шаблон в виде «Запрос-ответа», служба может передать отказ обратно без поля wsa:RelatesTo. Однако в некоторых случаях нет никакой гарантии, что отправитель может быть достигнут (например, если wsa:FaultTo содержит недействительный адрес, то нет никакого способа доставить отказ).

Во всех случаях служба может вернуться к правилам пункта 13.3, в соответствии с которыми не посылают никакой ответ. Служба может попытаться сохранить запросы в некоторый журнал, чтобы помочь опознать дефектного клиента.

14.5 Расширяемость отказов

Служба может включать дополнительную информацию об отказе сверх того, что определено в настоящем стандарте. Для этого предназначен расширяемый элемент s:Detail, в пределах которого может появиться любой XML при условии, что это должным образом отображено на пространство имен XML, которое определяет схему для этого содержимого. WS-Management использует эту технику расширения для значений URI wsman:FaultDetail, как показано в следующей схеме:

```
(1) <s:Detail>
(2) <wsman:FaultDetail>... </wsman:FaultDetail>
(3) <ExtensionData xmlns="пространство-имен-поставщика">...</ExtensionData>
(4) ...
(5) </s:Detail>
```

Элементы данных расширения могут появиться до или после любых расширений, специфичных для WS-Management и определенных данной спецификацией. Допустимо, если число элементов больше одного.

14.6 Основные отказы

Данный пункт включает все отказы из настоящего стандарта и всех нижележащих спецификаций. Данный список — нормативный список отказов для WS-Management.

R14.6-1 Служба должна возвратить отказы из следующего списка в случаях, когда операция, вызвавшая их, была сообщением из настоящего стандарта, для которого определены отказы. Служба, соответствующая настоящему стандарту, может возвратить другие отказы для сообщений, которые не являются частью WS-Management.

Для обеспечения совместимости клиентов важно, что в одинаковых ошибочных случаях используется один и тот же отказ. Если бы каждая служба возвращала свой собственный отказ для «Не найдено», построение интероперабельных клиентов было бы невозможно. В таблицах 5—43 исходная спецификация отказа основана на его имени QName.

Таблица 5 — wsman:AccessDenied

Подкод отказа	wsman:AccessDenied
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Отправитель не был уполномочен получить доступ к ресурсу
Дополнительные подробности	Нет
Замечания	Данный отказ возвращается в общем случае для всех отказов в доступе, которые относятся к ошибкам аутентификации или авторизации. Данный отказ не указывает на блокировку или конфликты при совместном выполнении или другие типы запретов, не связанных напрямую с защищенностью
Применимость	Любое сообщение
Обработка	Клиент указывает правильные идентификационные данные и повторяет операцию

Таблица 6 — wsa:ActionNotSupported

Подкод отказа	wsa:ActionNotSupported
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Действие не поддерживается службой
Дополнительные подробности	<s:Detail><wsa:Action> Неправильное URI действия </wsa:Action> </s:Detail> <!- если возможно, возвращается неподдерживаемое URI действия,->
Примечания	Данный отказ означает, что требуемое действие не поддерживается реализацией. Например, реализации только для чтения (поддерживающие только Get и Enumerate), возвращают данный отказ для любых операций кроме этих двух. Если реализация никогда не поддерживает действие, отказ может быть произведен как показано в поле «Дополнительные подробности» данной таблицы. Однако, если реализация в целом поддерживает действие, но ему не соответствует ресурс, дополнительные подробности могут быть расширены следующим образом: <s:Detail> <wsa:Action> Незаконное URI действия </wsa:Action> <wsman:FaultDetail> gost.e-ecolog.ru </wsman:FaultDetail> </s:Detail> Эта ситуация может произойти, например, когда реализация поддерживает Put, но клиент пытается обновить ресурс только для чтения
Применимость	Все сообщения
Обработка	Клиент определяет по метаданным, предоставленным службой, какие операции поддерживаются

Таблица 7 — wsman:AlreadyExists

Подкод отказа	wsman:AlreadyExists
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Отправитель попытался создать ресурс, который уже существует
Дополнительные подробности	Нет
Примечание	Данный отказ возвращается в случаях, когда пользователь попытался создать ресурс, который уже существует
Применимость	Create
Обработка	Клиент вызывает Put или создает ресурс с другим идентификатором

Таблица 8 — wsmen:CannotProcessFilter

Подкод отказа	wsmen:CannotProcessFilter
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Требуемый фильтр не может быть обработан
Дополнительные подробности	<s:Detail> <wsman:SupportedSelectorName>Корректное имя селектора для использования в выражении фильтра</wsman:SupportedSelectorName> * </s:Detail>
Примечания	Данный отказ возвращается для синтаксических ошибок или других семантических проблем с фильтром. Для использования с диалектом SelectorFilter (см. приложение Д), служба может включать один или несколько элементов SupportedSelectorName, чтобы предоставить список поддержанных имен селектора, если клиент запросил фильтрацию по одному или нескольким неподдержанным именам селекторов. Если фильтр корректен, но служба не может запустить фильтр из-за неверной конфигурации, отсутствия ресурсов или других внутренних проблем, более определенные отказы могут быть возвращены, такие как wsman:QuotaLimit или wsman:InternalError
Применимость	Enumerate
Обработка	Клиент исправляет проблему фильтра и пробует еще раз

Таблица 9 — wsman:CannotProcessFilter

Подкод отказа	wsman:CannotProcessFilter
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Требуемый фильтр не может быть обработан
Дополнительные подробности	<s:Detail> <wsman:SupportedSelectorName>Корректное имя селектора для использования в выражении фильтра</wsman:SupportedSelectorName> * </s:Detail>

Окончание таблицы 9

Примечания	Данный отказ возвращается для синтаксических ошибок или других семантических проблем с фильтром, таких как выход за границы подмножества, поддержанного службой. Для использования с диалектом SelectorFilter (см. Приложение Д), служба может включать один или несколько элементов SupportedSelectorName, чтобы предоставить список поддержанных имен селектора, если клиент запросил фильтрацию по одному или нескольким неподдержанным именам селекторов. Если фильтр корректен, но служба не может запустить фильтр из-за неверной конфигурации, отсутствия ресурсов или внутренних проблем, более определенные отказы могут быть возвращены, такие как wsman:QuotaLimit, wsman:InternalError, или wsme:EventSourceUnableToProcess
Применимость	Subscribe, операции доступа ресурса на уровне фрагмента
Обработка	Клиент исправляет проблему фильтра и пробует еще раз

Таблица 10 — wsman:Concurrency

Подкод отказа	wsman:Concurrency
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Действие не могло быть закончено из-за проблем блокировки или параллелизма
Дополнительные подробности	Нет
Примечания	Данный отказ означает, что требуемое действие не могло быть выполнено из-за внутреннего параллелизма или проблем блокировки или потому, что другой пользователь получает доступ к ресурсу. Данный отказ может произойти, когда перечисляется Enumerate и другой клиент пытается вызвать операции, такие как Delete, которые затронули бы результат происходящего перечисления
Применимость	Все сообщения
Обработка	Клиент ждет и пробует еще раз

Таблица 11 — wsme:DeliveryModeRequestedUnavailable

Подкод отказа	wsme:DeliveryModeRequestedUnavailable
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Требуемый режим доставки не поддерживается
Дополнительные подробности	<s:Detail> <wsme:SupportedDeliveryMode>...</wsme:SupportedDeliveryMode><wsme:SupportedDeliveryMode>...</wsme:SupportedDeliveryMode> ... </s:Detail> <! — Простой, необязательный список одного или нескольких URI поддерживаемых режимов доставки. Можно оставить пустым->
Примечания	Данный отказ возвращается для неподдерживаемых режимов доставки для указанного ресурса. Если стек поддерживает режим доставки в целом, но не для определенного ресурса, данный отказ может быть возвращен. Другие ресурсы могут поддерживать режим доставки. Отказ не подразумевает, что режим доставки не поддержан реализацией

Окончание таблицы 11

Применимость	Subscribe
Обработка	Клиент выбирает один из поддерживаемых режимов доставки

Таблица 12 — wsman:DeliveryRefused

Подкод отказа	wsman:DeliveryRefused
URI действия	gost.e-ecolog.ru
Код	s:Receiver
Причина	Получатель отказывается принимать доставку событий и просит, чтобы подписка была отменена
Дополнительные подробности	Нет
Примечания	Данный отказ возвращают получателю событий, чтобы вызвать отмену подписки. Данный отказ может произойти, когда клиент попытался отказаться от подписки сообщением Unsubscribe, но операция завершилась неудачно, или когда клиент потерял информацию об активных подписках и не хочет продолжать получать события, которыми больше не владеет. Данный отказ может помочь очистить список подписок в случаях, когда клиенты реконфигурированы или переустановлены, а их предыдущие подписки все еще активны
Применимость	Любое сообщение доставки событий в любом режиме
Обработка	Служба прекращает поставлять события для подписки и отменяет подписку, посылая подходящее сообщение SubscriptionEnd

Таблица 13 — wsa:DestinationUnreachable

Подкод отказа	wsa:DestinationUnreachable
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Нет маршрута к роли назначения, определенной заголовком адресации To
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> gost.e-ecolog.ru faultdetail>? </s:Detail> Когда используется модель адресации по умолчанию, поле wsman:FaultDetail может содержать gost.e-ecolog.ru 1/wsman/faultDetail/InvalidResourceURI
Примечания	Данный отказ возвращается как общий случай «Не Найдено» для ресурса, в котором ссылка оконечной точки ресурса не может быть соотнесена с реальным ресурсом. Данный отказ не используется для указания, что ресурс временно отключен. Для этого предназначен отказ wsa:EndpointUnavailable
Применимость	Все сообщения-запросы
Обработка	Клиент пытается определить версию службы, запросить любые метаданные и выполнить другие диагностические операции для определения причины, почему запрос не может быть доставлен

Таблица 14 — wsman:EncodingLimit

Подкод отказа	wsman:EncodingLimit
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Внутренний предел кодирования был превышен в запросе или может быть нарушен, если сообщение будет обработано
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> Опциональный; одна из величин, перечисленных ниже </wsman:FaultDetail> ... любое зависящее от службы дополнительное содержимое в формате XML... </s:Detail> Возможные значения в <wsman:FaultDetail>: Неподдерживаемый набор символов: gost.e-ecolog.ru Неподдержанный MTOM или другие типы кодировки: gost.e-ecolog.ru Запрошенный максимум был слишком большим. Запрошенный максимальный размер конверта был слишком маленьким. gost.e-ecolog.ru Слишком много опций: gost.e-ecolog.ru Возвращается, когда используется модель адресации по умолчанию и указывает, что слишком много селекторов использовались для соответствующего ResourceURI: gost.e-ecolog.ru Служба достигла своего собственного внутреннего предела при вычислении ответа: gost.e-ecolog.ru Операция завершилась успешно и не может быть отменена, но результат слишком большой для отправки gost.e-ecolog.ru Запрос содержал символ за пределами диапазона, который поддерживается службой: gost.e-ecolog.ru Слишком длинный URI: gost.e-ecolog.ru Не поддерживается использование пробела клиентской стороной: gost.e-ecolog.ru
Примечания	Данный отказ возвращается, когда превышен системный предел, опубликованный или специфичной службы
Применимость	Любые сообщения-запросы
Обработка	Клиент посылает сообщения, которые соответствуют пределам кодирования службы

Таблица 15 — wsa:EndpointUnavailable

Подкод отказа	wsa:EndpointUnavailable
URI действия	gost.e-ecolog.ru
Код	s:Receiver
Причина	Указанная точка назначения в настоящее время недоступна

Окончание таблицы 15

Дополнительные подробности	<s:Detail> <wsa:RetryAfter> xs:duration </wsa:RetryAfter><!-- Опционально-> ... необязательное определенное службой содержимое XML <wsman:FaultDetail>Детальное значение URI </wsman:FaultDetail> </s:Detail> gost.e-ecolog.ru Используется, когда ресурс известен, но временно недоступен
Примечания	Данный отказ возвращается, если сообщение было правильно и EPR был действителен, но указанный ресурс отключен. На практике службам трудно различать случаи «Не найдено» и «Отключено». В целом wsa:DestinationUnreachable предпочтителен
Применимость	Любые сообщения-запросы
Обработка	Клиент может повторить позже, после того, как ресурс будет снова подключен

Таблица 16 — wsman:EventDeliverToUnusable

Подкод отказа	wsman:EventDeliverToUnusable
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Источник события не может обработать подписку из-за того, что не может соединиться с оконечной точкой доставки событий, указанной в элементе Delivery
Дополнительные подробности	<s:Detail> ... любое специфичное для службы содержимое для определения ошибки... </s:Detail>
Примечания	Данный отказ ограничен случаями проблем установления соединения с адресом доставки. Эти проблемы включают: - адрес NotifyTo неприменим (система или устройство не достижимо, некорректно сформированный адрес, и т. д.). - разрешения не могут быть определены для доставки событий (например, wsman:Auth элемент не относится к поддержанному профилю безопасности, и т. д.). - идентификационные данные, связанные с NotifyTo, некорректны (например, пользовательский аккаунт не существует, отпечаток сертификата не является шестнадцатеричной строкой, и т. д.). Служба может включать дополнительную информацию, которая описывает ошибку соединения, чтобы помочь в решении проблемы соединения
Применимость	Subscribe
Обработка	Клиент гарантирует возможность соединения от компьютера службы назад к получателю событий, включая конфигурацию идентификации/разрешения и брандмауэров

Таблица 17 — wsme:EventSourceUnableToProcess

Подкод отказа	wsme:EventSourceUnableToProcess
URI действия	gost.e-ecolog.ru
Код	s:Receiver
Причина	Источник события не может обработать подписку
Дополнительные подробности	Нет

Окончание таблицы 17

Примечание	Данный источник событий не способен выполнить запрос Subscribe по внутренним причинам, не связанным с определенным запросом
Применимость	Subscribe
Обработка	Клиент повторяет подписку позже

Таблица 18 — wsmen:FilterDialectRequestedUnavailable

Подкод отказа	wsmen:FilterDialectRequestedUnavailable
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Требуемый диалект фильтрации не поддерживается
Дополнительные подробности	<s:Detail> <wsmen:SupportedDialect>....</wsmen:SupportedDialect> +</s:Detail>
Примечания	Данный отказ возвращается, когда клиент запрашивает тип фильтра или язык запросов, не поддержанный службой. Диалект фильтра может меняться от ресурса к ресурсу или может относиться ко всей службе
Применимость	Enumerate
Обработка	Клиент переключается на поддерживаемый диалект или выполняет простое перечисление без фильтра

Таблица 19 — wsme:FilteringNotSupported

Подкод отказа	wsme:FilteringNotSupported
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Фильтрация по источнику события не поддерживается
Дополнительные подробности	Нет
Примечание	Данный отказ возвращается, когда служба не поддерживает подписки с фильтрацией для указанного источника события и поддерживает только простую доставку всех событий для ресурса. Примечание — Служба может поддерживать фильтрацию по другому ресурсу событий или не поддерживает фильтрацию для любого ресурса. Применяется данный отказ
Применимость	Subscribe
Обработка	Клиент подписывается на доставку без фильтрации

Таблица 20 — wsmen:FilteringNotSupported

Подкод отказа	wsmen:FilteringNotSupported
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Фильтрованное перечисление не поддерживается
Дополнительные подробности	Нет

Окончание таблицы 20

Примечание	Данный отказ возвращается, когда служба вообще не поддерживает перечисления с фильтрацией, и поддерживает только простое перечисление. Если перечисление в целом не поддерживается, правильный отказ — wsa:ActionNotSupported. Примечание — Служба может поддерживать фильтрацию по другому ресурсу перечисления или не поддерживает фильтрацию для любого ресурса. Применяется данный отказ
Применимость	Enumerate
Обработка	Клиент переключается на простое перечисление

Таблица 21 — wsme:FilteringRequestedUnavailable

Подкод отказа	wsme:FilteringRequestedUnavailable
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Требуемый диалект фильтра не поддержан
Дополнительные подробности	<s:Detail> <wsme:SupportedDialect>..</wsme:SupportedDialect> + <wsman:FaultDetail>.. Если возможно, приведенное ниже значение URI</wsman:FaultDetail> </s:Detail> Возможное значение URI: gost.e-ecolog.ru
Примечания	Данный отказ возвращается, когда клиент запрашивает диалект фильтра, не поддерживаемый службой. В некоторых случаях подписка требует фильтр, так как результат нефильтрованной подписки может быть бесконечным или чрезвычайно большим. В этих случаях URI gost.e-ecolog.ru должен быть включен в элемент s:Detail
Применимость	Subscribe
Обработка	Клиент переключается на поддерживаемый диалект фильтра или не использует фильтрацию

Таблица 22 — wsman:FragmentDialectNotSupported

Подкод отказа	wsman:FragmentDialectNotSupported
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Требуемый язык или диалект фильтрации фрагмента не поддерживаются
Дополнительные подробности	<s:Detail> <wsman:FragmentDialect> xs:anyURI</wsman:fragmentdialect> <wsman:FragmentDialect> xs:anyURI</wsman:fragmentdialect> .... </s:Detail> Необязательные значения URI указывают поддерживаемые диалекты

Окончание таблицы 22

Примечания	Данный отказ возвращается, когда служба не поддерживает требуемый диалект фильтрации на уровне фрагмента. Если реализация поддерживает диалект фрагмента в целом, но не для определенного ресурса, также возвращается данный отказ. Другие ресурсы могут поддерживать диалект фрагмента. Данный отказ не подразумевает, что диалект фрагмента не поддерживается реализацией
Применимость	Enumerate, Get, Create, Put, Delete
Обработка	Клиент использует поддерживаемый диалект фильтрации или не использует фильтрацию

Таблица 23 — wsman:InternalError

Подкод отказа	wsman:InternalError
URI действия	gost.e-ecolog.ru
Код	s:Receiver
Причина	Служба не может выполнить запрос из-за внутренних ошибок обработки
Дополнительные подробности	<s:Detail> ... определенные для службы дополнительные элементы XML.... <s:Detail>
Примечания	Данный отказ — универсальная ошибка для фиксации внутренних ошибок обработки в внутри службы. Например, это правильный отказ, если служба не может загрузить необходимые исполнимые файлы, ее конфигурация испорчена, аппаратные средства не работают должным образом или происходят любые неизвестные или неожиданные внутренние ошибки. Ожидается, что служба должна повторно конфигурироваться, перезапускаться или переустанавливаться; таким образом, недостаточно, чтобы клиент повторил запрос
Применимость	Все сообщения
Обработка	Клиент восстанавливает службу средствами, не входящими в WS-Management

Таблица 24 — wsman:InvalidBookmark

Подкод отказа	wsman:InvalidBookmark
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Закладка, поставляемая подпиской, некорректна
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> Если возможно, одно из приведенных ниже значенийURI </wsman:FaultDetail> </s:Detail> Возможные значения URI: Служба не в состоянии вернуться к закладке и вопроизвести сообщения, начиная с указанной точки: http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/Expired Служба не в состоянии декодировать закладку: gost.e-ecolog.ru
Примечания	Данный отказ возвращается, если закладка истекла, искажена, или неизвестна

Окончание таблицы 24

Применимость	Subscribe
Обработка	Клиент отправляет новый запрос подписки без закладок или определяет правильное местонахождение закладки

Таблица 25 — wsmen:InvalidEnumerationContext

Подкод отказа	wsmen:InvalidEnumerationContext
URI действия	gost.e-ecolog.ru
Код	s:Receiver
Причина	Полученный контекст перечисления недействителен
Дополнительные подробности	Нет
Примечания	Недействительный контекст перечисления получен в сообщении. Как правило, данный отказ произойдет с Pull. Контекст перечисления может быть недействительным из-за истечения времени жизни, недопустимого формата или повторного использования старого контекста, больше не отслеживаемого службой. Служба также может возвратить данный отказ для любого случая, когда перечисление было завершено в одностороннем порядке на стороне службы, хотя желателен более конкретный отказ, потому что это обычно происходит из-за ошибок переполнения памяти (wsman:QuotaLimit), отказа авторизации (wsman:AccessDenied) или внутренних ошибок (wsman:InternalError)
Применимость	Pull, Release (в зависимости от типа подписки — по запросу или обычное перечисление)
Обработка	Клиент прекращает перечисление и позволяет службе удалить его по тайм-ауту, так как запрос Release тоже приведет к отказу

Таблица 26 — wsme:InvalidExpirationTime

Подкод отказа	wsme:InvalidExpirationTime
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Время срока окончания некорректно
Дополнительные подробности	Нет
Примечания	Время срока окончания некорректно вообще или в рамках службы. Данный отказ используется для прямых ошибок (окончание в прошлом, например) или срок окончания слишком далеко в будущем. Если служба не поддерживает срока окончания вообще, может быть возвращен отказ wsman:UnsupportedFeature с правильным детальным кодом
Применимость	Subscribe
Обработка	Клиент выпускает новую подписку с поддерживаемым временем срока окончания

Таблица 27 — wsmen:InvalidExpirationTime

Подкод отказа	wsmen:InvalidExpirationTime
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Время срока окончания некорректно

Окончание таблицы 27

Дополнительные подробности	Нет
Примечания	Поскольку WS-Management рекомендует не реализовывать функцию «Срок окончания», в большинстве реализаций данный отказ не может произойти. См. раздел 8 для получения дополнительной информации
Применимость	Enumerate
Обработка	Нет

Таблица 28 — wsme:InvalidMessage

Подкод отказа	wsme:InvalidMessage
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Сообщение-запрос имеет неизвестное или недействительное содержимое и не может быть обработано
Дополнительные подробности	Нет
Примечания	Данный отказ обычно не используется в WS-Management, хотя он может использоваться для случаев, не покрытых другими отказами. Если содержимое нарушает схему, можно послать отказ wsman:SchemaValidationError. Если определенные ошибки происходят в теле подписки, может использоваться один из более содержательных отказов. Данный отказ используется не для того, чтобы, указывать на неподдержанные характеристики, а только на неожиданное или неизвестное содержимое, нарушающее данную спецификацию
Применимость	Pub/sub запросы
Обработка	Клиент отправляет корректные сообщения, которые соответствуют настоящему стандарту

Таблица 29 — wsa:InvalidMessageInformationHeader

Подкод отказа	wsa:InvalidMessageInformationHeader
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Информационный заголовок сообщения некорректен, и сообщение не может быть обработано
Дополнительные подробности	<s:Detail> ... недействительный заголовок... </s:Detail>
Примечания	Данный отказ может произойти с любым типом ошибки заголовка SOAP. Заголовок может быть недействительным с точки зрения схемы или значения, или это может быть семантическая ошибка. Данный отказ не должен использоваться для указания на недействительный адрес ресурса (условие «не найден» для ресурса). Данный отказ указывает на фактические структурные нарушения правил заголовка SOAP в настоящем стандарте. Примеры — повторный MessageID, отсутствие RelatesTo в ответе, некорректно сформированные адреса или иное недостающее содержимое заголовка
Применимость	Все сообщения
Обработка	Клиент переформатирует сообщение, используя правильный формат, значения и число информационных заголовков в сообщении

Таблица 30 — wsman:InvalidOptions

Подкод отказа	wsman:InvalidOptions
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Одна или более опций некорректны
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> Если возможно, одно из приведенных ниже значений URI </wsman:FaultDetail> </s:Detail> Возможные значения URI: gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru
Примечания	Данный отказ в общем покрывает все случаи, когда имена опций или значения некорректны, или они используются в неправильных комбинациях
Применимость	Любые сообщения-запросы
Обработка	Клиент обнаруживает поддерживаемые имена опций и допустимые значения, запрашивая метаданными или пользуясь другими механизмами. Такие метаданные выходят за рамки настоящего стандарта

Таблица 31 — wsman:InvalidParameter

Подкод отказа	wsman:InvalidParameter
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Параметр операции некорректен
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> Если возможно, одно из приведенных ниже значений URI </wsman:FaultDetail> </s:Detail> Возможные значения URI: gost.e-ecolog.ru n/1/wsman/faultDetail/TypeMismatch gost.e-ecolog.ru
Примечания	Данный отказ возвращается, когда параметр специализированного действия некорректен. Данный отказ используется по умолчанию в новых реализациях, у которых должен быть универсальный отказ для этого случая. Метод может также возвратить любой определенный собственный отказ
Применимость	Все сообщения с действиями пользователя
Обработка	Клиент консультируется с WSDL операции и определяет корректный параметр

Таблица 32 — wsmt:InvalidRepresentation

Подкод отказа	wsmt:InvalidRepresentation
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Содержимое XML некорректно
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> Если возможно, одно из приведенных ниже значений URI </wsman:FaultDetail> </s:Detail> Возможные значения URI: gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru
Примечание	Данный отказ может быть возвращен, когда полученный XML некорректен семантически или использует неправильную схему для ресурса. Однако отказ wsman:SchemaValidationError может быть возвращен, если ошибка связана с нарушениями XML-схемы как таковыми, в противоположность недействительным семантическим значеням. Возможен аномальный случай, в котором не происходит нарушение схемы, но указано неправильное пространство имен; в этом случае возвращается gost.e-ecolog.ru
Применимость	Put, Create
Обработка	Клиент исправляет запрос XML

Таблица 33 — wsman:InvalidSelectors

Подкод отказа	wsman:InvalidSelectors
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Селекторы для ресурса некорректны
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> Если возможно, одно из приведенных ниже значений URI </wsman:FaultDetail> </s:Detail> Возможные значения URI: gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru
Примечание	Данный отказ покрывает все случаи, когда указанные селекторы были неправильными или неизвестными для указанного ресурса
Применимость	Любые сообщения-запросы
Обработка	Клиент проверяет документацию или метаданные и исправляет селекторы

Таблица 34 — wsa:MessageInformationHeaderRequired

Подкод отказа	wsa:MessageInformationHeaderRequired
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Обязательный заголовок отсутствует
Дополнительные подробности	<s:Detail> имя (XMLQName) недостающего заголовка </s:Detail>
Примечание	Обязательный информационный заголовок сообщения (To, MessageID или Action) не присутствует
Применимость	Все сообщения
Обработка	Клиент добавляет недостающий информационный заголовок сообщения

Таблица 35 — wsman:NoAck

Подкод отказа	wsman:NoAck
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Получатель не подтвердил доставку события
Дополнительные подробности	Нет
Примечание	Данный отказ возвращается, когда клиент (подписчик) получает событие с заголовком wsman:AckRequested и не отправляет (или не может отправить) подтверждение получения. Служба прекращает посылать события и завершает подписку
Применимость	Любое действие доставки событий (включая Heartbeat, пропущенные события и т. д.) в любом режиме доставки
Обработка	Для подписчиков: повторно запрашивается подписка без опции подтверждения. Для служб, предоставляющих события: служба немедленно отменяет подписку

Таблица 36 — wsman:QuotaLimit

Подкод отказа	wsman:QuotaLimit
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Служба занята, обслуживая другие запросы
Дополнительные подробности	Нет
Примечания	Данный отказ возвращается, когда сообщение SOAP корректно, но служба достигла предела ресурса или квоты
Применимость	Все сообщения
Обработка	Клиент может повторить позже

Таблица 37 — wsman:SchemaValidationError

Подкод отказа	wsman:SchemaValidationError
URI действия	gost.e-ecolog.ru

Окончание таблицы 37

Код	s:Sender
Причина	Поставляемый SOAP нарушает определение соответствующей схемы XML
Дополнительные подробности	Нет
Примечания	Данный отказ используется для ошибок синтаксического разбора XML или нарушений схемы. Полная валидация SOAP на соответствие схемам в режиме реального времени не ожидается, но процессоры могли бы фактически заметить нарушения схемы, такие как несоответствия типов. Во всех этих случаях применяется данный отказ. В режимах отладки, когда происходит валидация, данный отказ может быть возвращен для всех ошибок, обнаруженных валидирующим обработчиком
Применимость	Все сообщения
Обработка	Клиент исправляет сообщение

Таблица 38 — wsmen:TimedOut

Подкод отказа	wsmen:TimedOut
URI действия	gost.e-ecolog.ru
Код	s:Receiver
Причина	Время перечисления истекло. Перечисление более некорректно
Дополнительные подробности	Нет
Примечание	Данный отказ не должен использоваться в WS-Management из-за пересечения с wsman: TimedOut, который покрывает все другие сообщения
Применимость	Pull
Обработка	Клиент может повторить запрос Pull

Таблица 39 — wsman:TimedOut

Подкод отказа	wsman:TimedOut
URI действия	gost.e-ecolog.ru
Код	s:Receiver
Причина	Время выполнения операции истекло
Дополнительные подробности	Нет
Примечания	Операция не завершилась за время, указанное в значении wsman:OperationTimeout или переопределенном внутреннем таймауте, при обработке запроса. Данный отказ также возвращен во всех перечислениях, когда отсутствует содержимое, доступное для запроса Pull. Клиенты могут просто повторить запрос Pull до получения иного отказа
Применимость	Все запросы
Обработка	Клиент может повторить операцию. Если операция — запись (удаление, создание или иная операция), клиент может проверить журнал работы системы, прежде чем вслепую делать повторную попытку операции или выполнить Get или иную операцию чтения, чтобы попытаться обнаружить результат предыдущей операции

Таблица 40 — wsme:UnableToRenew

Подкод отказа	wsme:UnableToRenew
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Подписка не может быть возобновлена
Дополнительные подробности	Нет
Примечание	Данный отказ возвращается во всех случаях, когда подписка действительна, но не может быть возобновлена
Применимость	wsme:Renew
Обработка	Клиент запрашивает новую подписку

Таблица 41 — wsme:UnsupportedExpirationType

Подкод отказа	wsme:UnsupportedExpirationType
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Указанный тип срока окончания не поддерживается
Дополнительные подробности	Нет
Примечания	Определенное время для срока окончания (в противоположность продолжительности) не поддерживается. Данный отказ не должен использоваться, если само значение неправильно; отказ используется только если тип не поддерживается
Применимость	Subscribe
Обработка	Клиент исправляет срок окончания, используя продолжительность

Таблица 42 — wsmen:UnsupportedExpirationType

Подкод отказа	wsmen:UnsupportedExpirationType
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Указанный тип срока окончания не поддерживается
Дополнительные подробности	Нет
Примечания	Указанный срок окончания не поддерживается. Например, некоторый тип срока окончания, основанный на времени, не поддерживается (в противоположность типу срока окончания, основанному на продолжительности). Данный отказ не должен использоваться, если само значение неправильная; отказ используется только если тип не поддерживается
Применимость	Перечислить
Обработка	Клиент исправляет время срока окончания или опускает его и повторяет запрос

Таблица 43 — wsman:UnsupportedFeature

Подкод отказа	wsman:UnsupportedFeature
URI действия	gost.e-ecolog.ru

Окончание таблицы 43

Код	s:Sender
Причина	Указанная функция не поддерживается
Дополнительные подробности	<s:Detail> <wsman:FaultDetail> Если возможно, одно из приведенных ниже значений URI </wsman:FaultDetail> </s:Detail> Возможные значени URI: gost.e-ecolog.ru gost.e-ecolog.ru Mode gost.e-ecolog.ru Request gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru Mode gost.e-ecolog.ru gost.e-ecolog.ru Required gost.e-ecolog.ru Mismatch gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru gost.e-ecolog.ru Size gost.e-ecolog.ru gost.e-ecolog.ru
Примечание	Данный отказ указывает, что была запрошена неподдерживаемая функция
Применимость	Любое сообщение
Обработка	Клиент исправляет или удаляет запрос неподдерживаемых функций

Таблица 44 — wsme:UnsupportedExpirationType

Подкод отказа	wsme:UnsupportedExpirationType
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Поддерживается только продолжительность
Дополнительные подробности	Нет
Примечание	Данный отказ посылают, когда запрос Subscribe определяет срок окончания запроса, но источник событий способен принимать только продолжительность; например, если у источника события нет доступа к абсолютному времени

Окончание таблицы 44

Применимость	Subscribe, wsme:Renew
Обработка	Нет

Таблица 45 — wsmen:UnableToRenew

Подкод отказа	wsmen:UnableToRenew
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Текст, объясняющий причину отказа; например, «У источника события слишком много подписчиков»
Дополнительные подробности	Нет
Примечание	Данный отказ посылают, когда источник событий не способен к выполнению запроса Renew по внутренним причинам, не связанным с определенным запросом
Применимость	wsmen:Renew
Обработка	Нет

Таблица 46 — wsa:InvalidMessage

Подкод отказа	wsa:InvalidMessage
URI действия	gost.e-ecolog.ru
Код	s:Sender
Причина	Сообщение некорректно и не может быть обработано
Дополнительные подробности	Недействительное сообщение
Примечание	Если сообщение-запрос не соответствует схеме, то запрос должен завершиться неуспешно, и источник событий или менеджер подписки может вернуть данный отказ, указывающий, что запрос недействителен
Применимость	Subscribe, Renew, wsme:GetStatus, Unsubscribe
Обработка	Нет

Таблица 47 — wsme:CannotProcessFilter

Подкод отказа	wsme:CannotProcessFilter
URI действия	http://schemas.xmlsoap.org/ws/2004/08/addressing/fault
Код	s:Sender
Причина	Не может отфильтровать в соответствии с запросом
Дополнительные подробности	Нет
Примечание	Фильтр был определен, не может быть обработан
Применимость	Subscribe
Обработка	Нет

Приложение А
(справочное)

Условные обозначения

Данное приложение определяет нотацию и пространства имен, используемые в настоящем стандарте.

Настоящий стандарт использует следующий синтаксис для определения нормативных схем сообщений:

- синтаксис представлен как экземпляр XML, но значения, выделенные курсивом, указывают на типы данных вместо значений;

следующие символы добавляются к элементам и атрибутам для указания на число объектов:

- «?» (0 или 1);
- «*» (0 или больше);
- «+» (1 или больше);
- символ «|» указывает на выбор между альтернативами;
- символ «[и]» указывает, что приложенные элементы нужно рассматривать как группу относительно количества элементов или выбора;
- многоточие («...») указывает на точку расширения, в которую может быть добавлен иной дочерний элемент или атрибут. Дополнительные дочерние элементы и атрибуты могут быть добавлены в обозначенных точках расширения, но не должны противоречить семантике родителя или владельца, соответственно. Если получатель не признает расширение, получателю не следует обрабатывать сообщение, и можно направить сообщение об ошибке;
- префиксы пространства имен XML (см. таблицу А.1) указывают на пространство имен определяемого элемента.

Всюду по документу пробел в пределах значений элементов XML используется для удобочитаемости. На практике служба может принять и удалить начальные и завершающие пробелы в пределах значений элемента как будто пробел не использовался.

А.1 Пространства имен XML

Таблица А.1 перечисляет пространства имен XML, используемые в настоящем стандарте. Выбор любого префикса пространства имен произвольный и не имеет семантического значения. Если не указано иное, XML-схема для каждой спецификации может быть получена по адресу, равному XML URI для каждой спецификации, перечисленной в таблице А.1.

Таблица А.1 — Префиксы и пространства имен XML, используемые в настоящем стандарте

Префикс	Пространство имен XML	Спецификация
wsman	gost.e-ecolog.ru	*Данная спецификация*
wsmid	gost.e-ecolog.ru .org/wbem/wsman/identity/1/wsmanidentity.xsd	*Данная спецификация — определение версий поддерживаемого протокола*
s	gost.e-ecolog.ru	*SOAP 1.2*
xs	*http://www.w3.org/2001/XMLSchema*	*XML-схема 1*, *XML-схема 2*
wsdl	gost.e-ecolog.ru ap.org/wsdl	*WSDL/1.1*
wsa	*wsa04 или wsa10*	*Соответствует префиксу wsa04 либо wsa10*
wsa04	gost.e-ecolog.ru	*Пункт 5 настоящего стандарта*
wsa10	gost.e-ecolog.ru 05/08/addressing	*Рекомендация W3C WS-Addressing*
wsam	gost.e-ecolog.ru 07/05/addressing/metadata	*Рекомендация W3C Метаданные WS-Addressing*
wsme	gost.e-ecolog.ru	*Раздел 10*
wsmen	gost.e-ecolog.ru	*Раздел 8*
wsmt	gost.e-ecolog.ru	*Раздел 7*
wsp	gost.e-ecolog.ru	*WS-Policy*

Приложение В
(обязательное)

Соответствие

Данное приложение определяет правила соответствия, используемые в настоящем стандарте.

Реализация не соответствует настоящему стандарту, если она не удовлетворяет хотя бы одному обязательному тебованию, определенному в правилах соответствия для каждой секции и обозначаемому следующим образом:

R*nnnn*: Текст правила

Общие правила соответствия определены следующим образом:

RB-1 Чтобы соответствовать стандарту, служба должна выполнять все правила, определенные в настоящем стандарте. Элементы, отмеченные ключевым словом «должно», обязательны для реализации, и элементы, отмеченные ключевым словом «следует», настоятельно рекомендуются к реализации для достижения максимальной степени функциональной совместимости. Элементы, отмеченные ключевым словом «может», обозначают предпочтительную реализацию для достижения ожидаемых характеристик, но функциональная совместимость не пострадает, если они проигнорированы.

RB-2 Службы, соответствующие настоящему стандарту, должны использовать следующий URI пространства имен XML:

(1) hgost.e-ecolog.ru

RB-3 Узел SOAP не должен использовать идентификатор пространства имен XML настоящего стандарта, если он не удовлетворяет требованиям соответствия настоящего стандарта.

Настоящий стандарт не требует, чтобы поддерживались все сообщения и операции. Настоящий стандарт только требует, чтобы любое поддерживаемое сообщение или операция следовали установленным для них правилам соответствия. Важно, чтобы службы не использовали идентификатор пространства имен XML для WS-Management в операциях SOAP, если они не совместимы с правилами, определенными в настоящем стандарте.

Приложение С
(обязательное)

HTTP (S) транспорт и профиль безопасности

С.1 Общие положения

Хотя WS-Management является протоколом SOAP и не связан с определенным сетевым транспортом, для функциональной совместимости требуется, чтобы были установлены некоторые единые стандарты. Данный раздел связан по установлению общего использования протоколов HTTP 1.1 и HTTPS. В дополнение к HTTP и HTTPS, настоящий стандарт позволяет использовать в качестве транспорта для сообщений WS-Management любой транспортный протокол, поддерживающий SOAP.

Для целей идентификации и ссылки каждый транспорт идентифицируется URI, и каждый механизм аутентификации, определенный в настоящем стандарте, также идентифицируется URI.

По мере стандартизации новых транспортных средств они могут также получить URI для ссылки, и любым новым механизмам аутентификации, которые они предоставляют, можно также назначить URI для целей публикации и идентификации в XML-документах. По мере стандартизации новых транспортных средств для WS-Management соответствующие специальные требования могут быть определены и изданы, чтобы гарантировать функциональную совместимость.

Для функциональной совместимости стандартными транспортными средствами являются HTTP 1.1 (RFC 2616) и HTTPS (использующий TLS 1.0) (RFC 2818).

Привязка SOAP к HTTP, описанная в разделе 7 Часть 2 SOAP версии 1.2: дополнения используется для кодирования WS-Management для передачи по HTTP и HTTPS.

С.2 Привязка к HTTP (S)

Данный раздел разъясняет, как сообщения SOAP связаны с HTTP (S).

RC.2-1 Служба, которая поддерживает привязку SOAP к HTTP (S), должна, по крайней мере, поддерживать HTTP 1.1.

RC.2-2 Служба должна, по крайней мере, реализовать отвечающий узел SOAP шаблона обмена сообщениями «Запрос-ответ» SOAP:

gost.e-ecolog.ru

RC.2-3 Служба может не реализовывать отвечающий узел SOAP шаблона обмена сообщениями «Ответ» SOAP:

gost.e-ecolog.ru)03/05/soap/mep/soap-response/

RC.2-4 Служба может не поддерживать функцию SOAP «Веб-метод».

RC.2-5 Служба должна, по крайней мере, релизовать отвечающий узел SOAP одностороннего обмена HTTP, в котором конверт SOAP передается в запросе HTTP, а в ответе HTTP передается код ответа 202 Accepted и пустое тело (без конверта SOAP).

Шаблон обмена сообщениями, упомянутый в RC.2-5, используется для передачи сообщений SOAP, которые не требуют никакого ответа.

RC.2-6 Служба должна поддерживать передачу конвертов SOAP и одностороннюю доставку конверта SOAP с использованием HTTP Post.

RC.2-7 В случаях, когда служба не может ответить сообщением SOAP, следует возвратить код 500 ошибки HTTP (внутренняя ошибка сервера), и клиентской стороне следует закрыть соединение.

RC.2-8 В случае, когда служба поддерживает HTTPS (TLS 1.0), служба должна реализовать как минимум TLS_RSA_WITH_RC4_128_SHA. Рекомендуется, чтобы служба также поддерживала TLS_RSA_WITH_AES_128_CBC_SHA.

RC.2-9 При доставке отказов следует использовать код состояния HTTP 500 для отказов s:Receiver и следует использовать код 400 для отказов s:Sender.

RC.2-10 Не требуется, чтобы URL, используемый для для передачи сообщение SOAP по методу HTTP-Post, имел то же самое содержимое как URI из wsa:To, используемый в адресе SOAP. Часто URL HTTP имеет то же самое содержимое как URI из wsa:To в сообщении, но может дополнительно содержать другие поля маршрутизации сообщений, добавленные как суфикс к сетевому адресу, используя определенную службой последовательность символов разделителя. Рекомендуется, чтобы службы потребовали сетевой адрес wsa:To URL только для унификации обработкии поведения клиентской стороны и включали маршрутизацию на уровне службы в другие части адреса.

RC.2-11 В отсутствие других требований рекомендуется, чтобы в случае использования HTTP-POST часть URL, определяющая путь, была /wsman для ресурсов, которые требуют аутентификацию, и /wsman-anon для ресурсов, которые не требуют аутентификацию. Если эти пути используются, неаутентифицированные запросы не должны поддерживаться для /wsman, и аутентификация не должна требоваться для /wsman-anon.

RC.2-12 Если заголовок SOAP Action присутствует в запросе HTTP/HTTPS, который несет сообщение SOAP, он должен соответствовать URI wsa:Action из этого сообщения SOAP. Заголовок SOAP Action необязательный, и служба не должна отклонять запрос, если данный заголовок отсутствует.

Поскольку WS-Management основан на SOAP 1.2, необязательный заголовок SOAP Action просто используется в качестве оптимизации. Если этот заголовок присутствует, он должен соответствовать URI wsa:Action, используемому в сообщении SOAP. Служба может отклонить запрос при инспекции заголовка SOAPAction без обработки содержимого SOAP, если действие некорректно. Однако служба не может отклонить запрос, если заголовок SOAP Action опущен.

RC.2-13 Если служба поддерживает вложения, то служба должна поддерживать функцию оптимизации передачи HTTP.

RC.2-14 Если служба не может обработать сообщение с вложением или неподдерживаемым типом кодировки и в качестве транспорта используется HTTP или HTTPS, то служба должна возвратить ошибку HTTP 415 (неподдерживаемый тип данных).

RC.2-15 Если служба не может обработать сообщение с вложением или неподдерживаемым типом кодировки при использовании транспортных средств, отличных от HTTP/HTTPS, то следует вернуть сообщение-отказ wsman:EncodingLimit со следующим кодом детализации:

gost.e-ecolog.ru

С.3 Профили безопасности HTTP (S)

Настоящий стандарт определяет ряд профилей безопасности для использования с HTTP и HTTPS. Службы, соответствующие настоящему стандарту, не обязаны поддерживать HTTP или HTTPS, но, если поддерживают, эти предопределенные профили предоставляют клиенту по крайней мере один способ получить доступ к службе. Другие спецификации могут определить дополнительные профили для использования с HTTP или HTTPS.

RC.3-1 Служба, соответствующая настоящему стандарту, поддерживающая HTTP, должна поддерживать один из предопределенных профилей HTTP.

RC.3-2 Служба, соответствующая настоящему стандарту, поддерживающая HTTPS, должна поддерживать один из предопределенных профилей HTTPS.

RC.3-3 Службе, соответствующей настоящему стандарту, не следует предоставлять WS-Management по полностью неаутентифицированному каналу HTTP, за исключением Identify (см. раздел 11), отладки, или в случаях, определенных службой.

Служба не обязана экспортировать только единственный адрес HTTP или HTTPS. Служба может иметь несколько адресов, каждый из которых поддерживает определенный профиль безопасности или несколько профилей.

Если клиенты поддерживают все предопределенные профили, они уверены в некоторой форме безопасного доступа к WS-Management, которая поддерживает HTTP, HTTPS или оба протокола.

С.3.1 gost.e-ecolog.ru

Данный профиль — это, по существу, «стандартный» профиль, ограниченный базовой аутентификацией.

Типичная последовательность показана в таблице С.1.

Таблица С.1 — Последовательность стандартной аутентификации

	Клиент		Служба
1	Клиент соединяется без заголовка аутентификации	è	Служба не видит заголовка
2		ç	Служба посылает код возврата 401, указывая Basic как метод аутентификации
3	Клиент предоставляет заголовок аутентификации Basic	è	Служба подтверждает подлинность клиента

Такое поведение нормально для HTTP. Если клиент устанавливает соединение с корректным заголовком авторизации Basic, запрос обрабатывается немедленно.

Стандартная аутентификация не рекомендуется для незащищенных транспортных каналов. Если используется с одним только HTTP, то, например, передача пароля составляет угрозу безопасности. Однако, если транспорт HTTP защищен, например, IPSec, риск существенно снижен.

Аналогично базовая аутентификация подходит для тестирования, прототипирования или диагностики.

С.3.2 gost.e-ecolog.ru

Данный профиль — это, по существу, «стандартный» профиль, ограниченный использованием дайджест-аутентификации.

Типичная последовательность показана в таблице С.2.

Таблица С.2 — Последовательность дайджест-аутентификации

	Клиент		Служба
1	Клиент соединяется без заголовка аутентификации	è	Служба не видит заголовка
2		ç	Служба посылает код возврата 401, указывая Digest как метод аутентификации
3	Клиент предоставляет заголовок аутентификации Digest	è	
4		ç	Служба сразу в ответе 401 возвращает необходимые токены (realm, nonce, qop)
5	Клиент продолжает последовательность авторизации	è	Служба подтверждает подлинность клиента

Такое поведение нормально для HTTP.

С.3.3 gost.e-ecolog.ru

Данный профиль устанавливает использование базовой аутентификации по HTTPS. Данный профиль используется, когда соединение шифруется с использованием сертификата серверной стороны, но служба должна подтвердить подлинность клиента.

Типичная последовательность приведена в таблице С.3.

Таблица С.3 — Последовательность стандартной аутентификации по HTTPS

	Клиент		Служба
1	Клиент соединяется без заголовка аутентификации, используя HTTPS	è	Служба не видит заголовка, но устанавливает зашифрованное соединение
2		ç	Служба посылает код возврата 401, указывая Basic как метод аутентификации
3	Клиент предоставляет заголовок аутентификации Basic	è	Служба подтверждает подлинность клиента

Если клиент устанавливает соединение с корректным заголовком авторизации Basic, запрос обрабатывается немедленно.

С.3.4 gost.e-ecolog.ru

Данный профиль устанавливает использование аутентификации Digest по HTTPS. Данный профиль используется, когда соединение шифруется с использованием сертификата серверной стороны, но служба должна подтвердить подлинность клиента.

Типичная последовательность приведена в таблице С.4.

Таблица С.4 — Последовательность аутентификации Digest по HTTPS

	Клиент		Служба
1	Клиент соединяется без заголовка аутентификации, используя HTTPS	è	Служба не видит заголовка
2		ç	Служба посылает код возврата 401, указывая Digest как метод аутентификации
3	Клиент предоставляет заголовок аутентификации Digest	è	
4		ç	Служба сразу в ответе 401 возвращает необходимые токены (realm, nonce, qop)
5	Клиент продолжает последовательность авторизации	è	Служба подтверждает подлинность клиента

Это поведение нормально для HTTPS. В схеме дайджест-аутентификации HTTP не предполагается Digest запрос от клиента.

С.3.5 gost.e-ecolog.ru

В этом режиме безопасности клиент предоставляет сертификат X.509 для подтверждения подлинности клиента. Никакой заголовок аутентификации HTTP или HTTPS не требуется в запросе HTTP-POST.

Однако как подсказка службе может использоваться следующий заголовок аутентификации HTTP/HTTPS:

Authorization:

gost.e-ecolog.ru

Так как служба может быть настроена таким образом, чтобы всегда запрашивать сертификат, данный заголовок аутентификации необязателен.

Эта простая последовательность приведена в таблице С.5.

Таблица С.5 — Последовательность HTTPS с сертификатом клиента

	Клиент		Служба
1	Клиент соединяется без заголовка аутентификации, но представляет сертификат X.509	è	Служба игнорирует заголовок аутентификации и извлекает сертификат клиентской стороны, использованный в рукопожатии TLS 1.0
2		ç	Служба предоставляет доступ или отклоняет запрос с кодом возврата 403.7 или 403.16

С.3.6 gost.e-ecolog.ru

В этом профиле сначала используется профиль gost.e-ecolog.ru https/mutual, чтобы подтвердить подлинность обеих сторон, используя сертификаты X.509. Последующие отдельные операции аутентифицированы заголовками базовой авторизации HTTP.

Данный профиль подтверждает подлинность и клиента, и службы и обеспечивает один уровень защищенности, как правило на уровне компьютера или устройства. Второй уровень идентификации, как правило, выполняет авторизацию для определенных операций, хотя может действовать как простой, вторичный механизм аутентификации без семантики авторизации. Типичная последовательность приведена в таблице С.6.

Таблица С.6 — Последовательность базовой аутентификации по HTTPS с сертификатом клиента

	Клиент		Служба
1	Клиент соединяется с сертификатом и специальным заголовком авторизации	è	Служба запрашивает сертификат клиента и подтверждает подлинность. Если сертификат отсутствует или некорректен, последовательность останавливается здесь с кодами возврата 403.7 или 403.16
2		ç	После подтверждения сертификата служба посылает код возврата 401, указывая требование предоставить заголовок авторизации Basic
3	Клиент выбирает Basic как режим аутентификации и включает его в заголовок аутентификации, как определено для HTTP 1.1	è	Служба вновь подтверждает подлинность клиента прежде, чем выполнить операцию

В начальном запросе заголовок аутентификации HTTPS должен быть следующим:

Authorization:

gost.e-ecolog.ru

Это указывает службе, что используется данный особый способ, и что служба может запросить сертификат клиента, чтобы гарантировать что на последующие запросы необходимо направлять вызов (challenge) для базовой авторизации, если заголовок HTTP Authorization отсутствует в запросе.

Заголовок Authorization интерпретируется как обычный базовый доступ HTTP:

Authorization: Basic ...закодированные имя_пользователя/пароль

Такое использование стандартной аутентификации безопасно (в отличие от его типичной эксплуатации в HTTP), потому что передача имени пользователя и пароля выполнена по зашифрованному соединению TLS 1.0.

С.3.7 gost.e-ecolog.ru

Данный профиль совпадает с gost.e-ecolog.ru , за исключением того, что использует модель дайджест-аутентификации HTTP после начальной взаимной аутентификации, основанной на сертификатах X.509.

В начальном запросе заголовок аутентификации HTTPS должен быть следующий:

Authorization:

gost.e-ecolog.ru

С.3.8 gost.e-ecolog.ru

В этом профиле клиент соединяется с сервером, используя HTTPS только с сертификатом серверной стороны, чтобы зашифровать связь.

Идентификация основывается на RFC 4559, который описывает использование GSSAPI SPNEGO по HTTP (таблица С.7). Данный механизм позволяет HTTP выполнять протокол согласования RFC 4178 для подтверждения подлинности пользователя, основываясь на версии 5 Kerberos.

Таблица С.7 — Последовательность идентификации SPNEGO по HTTPS

	Клиент		Служба
1	Клиент соединяется без заголовка аутентификации, используя HTTPS	è	Служба не видит заголовка, но устанавливает зашифрованное соединение
2		ç	Служба посылает код возврата 401, указывая **Negotiate** как доступный механизм аутентификации HTTP
3	Клиент использует указанный выше PROPOSED STANDARD, чтобы начать последовательность SPNEGO, для согласования по Kerberos V5	è	...
4	...	ç	Служба участвует в последовательности SPNEGO, чтобы подтвердить подлинность клиента, использующего Kerberos V5
5	Клиент аутентифицирован	è	Служба подтверждает подлинность клиента

С.3.9 gost.e-ecolog.ru

Данный способ совпадает с gost.e-ecolog.ru secprofile/https/spnego-kerberos за исключением того, что сервер и клиент взаимно подтверждают подлинность друг друга на уровне TLS до начала последовательности идентификации Kerberos (таблица С.8). Подробнее см. RFC 4178.

Таблица С.8 — Последовательность идентификации SPNEGO по HTTPS с сертификатом клиента

	Клиент		Служба
1	Клиент соединяется без заголовка аутентификации, используя HTTPS	è	Служба запрашивает сертификат клиента и подтверждает подлинность. Если сертификат отсутствует или некорректен, последовательность останавливается здесь с кодом возврата 403.7 или 403.16
2		ç	После взаимной идентификации с использованием сертификатов служба посылает код возврата 401, указывая **Negotiate** как доступный механизм аутентификации HTTP
3	Клиент использует PROPOSED STANDARD, чтобы начать последовательность SPNEGO, для согласования по Kerberos V5	è	...
4	...	ç	Служба участвует в последовательности SPNEGO, чтобы подтвердить подлинность клиента, использующего Kerberos V5
5	Клиент аутентифицирован	è	Служба подтверждает подлинность клиента

Как правило, это используется для того, чтобы взаимно подтвердить подлинность устройств или компьютеров, и затем впоследствии аутентифицировать пользователя или роль.

C.3.10 gost.e-ecolog.ru

Данный профиль совпадает с gost.e-ecolog.ru secprofile/https/spnego-kerberos за исключением того, что он выполнен по соединению HTTP. Подробнее см. RFC 4178.

Несмотря на то, что данный профиль поддерживает безопасную аутентификацию, отсутствие шифрования представляет угрозы безопасности, такие как раскрытие информации, так как сообщения SOAP передаются открытым текстом. Это не должно использоваться в окружениях, которые требуют высокого уровня защищенности.

C.4 IPSec и HTTP

HTTP с базовой аутентификацией слаб по отношению к атакам в небезопасной сети. Однако использование IPSec может решить эту проблему. IPSec обеспечивает высококачественную криптографическую защищенность, аутентификацию источников данных и средства защиты от повторного воспроизведения.

Поскольку IPSec предназначен для аутентификации компьютеров и защиты сетевого трафика, этого недостаточно для реального менеджмента во многих случаях, когда может потребоваться дополнительная идентификация определенных пользователей для доступа к ресурсам. IPSec должен использоваться вместе с одним из профилей из данного пункта для идентификации пользователя. Однако это устраняет потребность в HTTPS и позволяет безопасное использование профилей, основанных на HTTP.

С сетевой точки зрения использование базовой аутентификации HTTP по каналу, обеспеченному IPSec, безусловно безопасно и эквивалентно использованию HTTPS с сертификатами серверной стороны. Например, профиль безопасности gost.e-ecolog.ru (использующий HTTPS) эквивалентен простому gost.e-ecolog.ru (использующий HTTP), если защита трафика обеспечена IPSec.

Другие спецификации могут определить профили безопасности IPsec, которые объединяют IPSec с соответствующими механизмами аутентификации.

Приложение D
(справочное)

Поддержка XPath

D.1 Общие положения

Реализации, как правило, должны поддерживать XPath для нескольких целей, таких как доступ на уровне фрагмента (см. 7.7), наборы данных (см. раздел 8), и фильтрация (см. 10.2.2). Поскольку полная спецификация XPath 1.0 большая, то, как правило, в реализациях с ограниченными ресурсами требуются подмножества.

Цель — определить минимальный набор синтаксических элементов, которые могут обеспечить реализации максимальную функциональную совместимость (интероперабельность). В большинстве случаев реализации поддерживают большие подмножества полного XPath, но им нужны дополнительные определения, чтобы гарантировать, что подмножества отвечают минимальным требованиям. Определения BNF уровней 1 и 2 в этом приложении устанавливают такие минимумы для использования в пространстве WS-Management.

Настоящий стандарт определяет два профиля подмножества для XPath: Уровень 1 с базовой поддержкой селектора узла и без фильтрации (для поддержки доступа на уровне фрагмента, как описано в 7.7), и уровень 2 с базовой поддержкой фильтрации (для перечисления и получения уведомлений). Уровень 2 является формальным над множеством уровня 1.

Следующие BNF являются формальными LL (1) грамматиками. Анализатор может быть построен автоматически из BNF с использованием соответствующего инструмента, или рекурсивный нисходящий синтаксический анализатор может быть реализован вручную по грамматике.

В тексте грамматик нетерминальные символы окружены угловыми скобками, и терминальные символы находятся в верхнем регистре и не окружены угловыми скобками.

Поддержка пространства имен XML явно отсутствует в этих определениях. Процессоры, которые отвечают требованиям синтаксиса, могут обеспечить режим, в котором элементы обрабатываются без отношения к пространствам имен XML, но могут также обеспечить режим с поддержкой пространств имен.

Контекст выполнения по умолчанию XPath определен явно в 8.4 и 10.2.2.

Ожидается, что для следующих диалектов пространства имен XML и квалифицированные имена (QName) не будут поддержаны по умолчанию и могут быть тихо проигнорированы реализацией.

Эти диалекты приведены исключительно для информационных целей и не предназначены для использования как диалекты фильтра в фактических сообщениях SOAP. Поскольку они соответствуют XPath (несмотря на то, что являются подмножествами), диалект фильтра в сообщениях SOAP является диалектом полного XPath:

gost.e-ecolog.ru Ƨ/1999/REC-xpath-19991116

D.2 Уровень 1

Уровень 1 содержит минимально необходимый XPath для определения узлов в XML-документе или фрагменте и предназначен для использования с доступом на уровне фрагмента (см. 7.7).

Пример:

(1) <path> ::= <root_selector> TOKEN_END_OF_INPUT;

(2) <root_selector> ::= TOKEN_SLASH <element_sequence>;

(3) <root_selector> ::= <attribute>;

(4) <root_selector> ::= <relpath><element_sequence>;

(5) <root_selector> ::= TOKEN_DOT;

(6) <relpath> ::= <>;

(7) <relpath> ::= TOKEN_DOT TOKEN_SLASH;

(8) <relpath> ::= TOKEN_DOT_DOT TOKEN_SLASH;

(9) <element_sequence> ::= <element><optional_filter_expression><more>;

(10) <more> ::= TOKEN_SLASH <follower>;

(11) <more> ::= <>;

(12) <follower> ::= <attribute>;

(13) <follower> ::= <text_function>;

(14) <follower> ::= <element_sequence>;

(15) <optional_filter_expression> ::=

(16) TOKEN_OPEN_BRACKET <filter_expression> TOKEN_CLOSE_BRACKET;

(17) <optional_filter_expression> ::= <>;

(18) <attribute> ::= TOKEN_AT_SYMBOL <name>;

(19) <element> ::= <name>;

(20) <text_function> ::=

(21) TOKEN_TEXT TOKEN_OPEN_PAREN TOKEN_CLOSE_PAREN;

(22) <name> ::= TOKEN_XML_NAME;

(23) <filter_expression> ::= <array_location>;

(24) <array_location> ::= TOKEN_UNSIGNED_POSITIVE_INTEGER

Данный диалект позволяет выбирать любой узел XML по имени или положению в массиве или любой атрибут по имени. Дополнительно тест для узла text() может завершать выражение, чтобы выбирать только значение имени, исключая обертку имени элемента XML.

Терминалы в грамматике определены как показано в таблице D.1.

Таблица D.1 — Терминалы уровня 1 XPath

TOKEN_SLASH	Символ '/'
TOKEN_DOT	Символ '.'
TOKEN_DOT_DOT	Символы '..'
TOKEN_END_OF_INPUT	Конец входной строки
TOKEN_OPEN_BRACKET	Символ '['
TOKEN_CLOSE_BRACKET	Символ ']'
TOKEN_AT_SYMBOL	Символ '@'
TOKEN_XML_NAME	Эквивалент типа xs:token схемы XML
TOKEN_UNSIGNED_POSITIVE_INTEGER	Значение в диапазоне 1.. 4294967295
TOKEN_TEXT	Символы 'text'
TOKEN_OPEN_PAREN	Символ '('
TOKEN_CLOSE_PAREN	Символ ')'

В последующих примерах используется следующий фрагмент XML. Предполагается, что узел "a" выступает в качестве контекста (то есть представляет ресурс или документ события).

Пример 1:

(1) <Envelope>

(2) <Body>

(3) <a>

*(4) 100 *

(5) <c>

(6) <d> 200 </d>

(7) </c>

(8) <c>

(9) <d> 300 </d>

(10) <d> 400 </d>

(11) </c>

*(12) *

(13) </Body>

(14) </Envolope>

Пример 2:

(1) / // Выбирает <a> и все его содержимое

(2) /a // Выбирает <a> и все его содержимое

(3).// Выбирает <a> и все его содержимое

(4) ../a // Выбирает <a> и все его содержимое

(5) b // Выбирает <bx ="y">100

(6) c //Выбирает оба узла <c>, один за другим

(7) c[1] // Выбирает <c><d>200</d></c>

(8) c[2]/d[2] // Выбирает <c><d>400</d></c>

(9) c[2]/d[2]/text() // Выбирает 400

(10) b/text() //Выбирает 100

(11) b/x // Выбирает x="y"

Единственная возможность фильтрации — это выбор массива. XPath может возвратить множество узлов. Подраздел 7.7 предполагает выбор отдельного узла, а не множества узлов, поэтому если возникнет ситуация, проиллюстрированная выше на строке (6), большинство реализаций просто возвращают отказ с сообщением о том, что неясно, который <c> предназначался, и требованием, чтобы клиент фактически выбрал один из доступных двух элементов <c>, используя синтаксис массива. Кроме того, к выбору атрибута нельзя приписать text().

Службам, которые поддерживают доступ на уровне фрагмента, как описано в 7.7, рекомендуется поддерживать подмножество XPath, по крайней мере, столь же сильное, как это описано на уровне 1.

Очевидно, служба может предоставить полный XPath 1.0 или любое другое подмножество, которое соответствует или превышает требования, определенные в данном пункте.

Служба, поддерживающая диалект XPath уровень 1, должна обеспечить выделение отдельного узла. Если на одном уровне XML есть более чем один элемент с одинаковым именем, для их различения должна использоваться нотация массива.

D.3 Уровень 2

Уровень 2 содержит все, определенное на уровне 1, плюс функциональность фильтрации общего назначения со стандартным набором операторов отношения и подвыражений в скобках (с И, ИЛИ, НЕ и так далее). Данный диалект подходит для фильтрации перечислений и подписки. Данный диалект — строгое надмножество Уровня 1. В нем существенно расширена продукция <filter_expression>, чтобы содержать полезное подмножество синтаксиса фильтрации XPath.

Пример:

(1) <path> ::= <root_selector> TOKEN_END_OF_INPUT;

(2) <root_selector> ::= TOKEN_SLASH <element_sequence>;

(3) <root_selector> ::= <relpath><element_sequence>;

(4) <root_selector> ::= <attribute>;

(5) <root_selector> ::= TOKEN_DOT;

(6) <relpath> ::= <> ;

(7) <relpath> ::= TOKEN_DOT TOKEN_SLASH;

(8) <relpath> ::= TOKEN_DOT_DOT TOKEN_SLASH;

(9) <element_sequence> ::= <element><optional_filter_expression><more>;

(10) <more> ::= TOKEN_SLASH <follower>;

(11) <more> ::= <>;

(12) <follower> ::= <attribute>;

(13) <follower> ::= <text_function>;

(14) <follower> ::= <element_sequence>;

(15) <optional_filter_expression> ::= TOKEN_OPEN_BRACKET <filter_expression>
0936 TOKEN_CLOSE_BRACKET;

(16) <optional_filter_expression> ::= <>;

(17) <attribute> ::= TOKEN_AT_SYMBOL <name>;

(18) <element> ::= <name>;

(19) <text_function> ::= TOKEN_TEXT TOKEN_OPEN_PAREN TOKEN_CLOSE_PAREN;

(20) <name> ::= TOKEN_XML_NAME;

(21) <filter_expression> ::= <array_location>;

(22) <array_location> ::= TOKEN_UNSIGNED_POSITIVE_INTEGER;

(23) // Next level, simple OR expression

(24) <or_expression> ::= <and_expression><or_expression_rest>;

(25) <or_expression_rest> ::= TOKEN_OR <and_expression><or_expression_rest>;
(26) <or_expression_rest> ::= <>;
(27) // Next highest level, AND expression
(28) <and_expression> ::= <rel_expression><and_expression_rest>;
(29) <and_expression_rest> ::= TOKEN_AND <rel_expression>
<and_expression_rest>;
(30) <and_expression_rest> ::= <>;
(31) // Next level of precedence >, <, >=, <=, =, !=
(32) <rel_expression> ::= <sub_expression><rel_expression_rest>;
(33) <rel_expression_rest> ::= <name><rel_op><const>;
(34) <rel_expression_rest> ::= <>;
(35) // Identifier, literal, or identifier + param_list (function call)
(36) <sub_expression> ::= TOKEN_OPEN_PAREN <filter_expression>
TOKEN_CLOSE_PAREN;
(37) <sub_expression> ::= TOKEN_NOT TOKEN_OPEN_PAREN <filter_expression>
TOKEN_CLOSE_PAREN;
(38) // Relational operators
(39) <rel_op> ::= TOKEN_GT; // >
(40) <rel_op> ::= TOKEN_LT; // <
(41) <rel_op> ::= TOKEN_GE; // >=
(42) <rel_op> ::= TOKEN_LE; // <=
(43) <rel_op> ::= TOKEN_EQ; // =
(44) <rel_op> ::= TOKEN_NE; // !=
(45) <const> ::= QUOTE TOKEN_STRING QUOTE.

Терминалы в грамматике определены, как показано в таблице D.2.

Таблица D.2 — Терминалы уровня 2 XPath

TOKEN_SLASH	Символ '/'
TOKEN_DOT	Символ '.'
TOKEN_DOT_DOT	Символы '..'
TOKEN_END_OF_INPUT	Конец входной строки
TOKEN_OPEN_BRACKET	Символ '['
TOKEN_CLOSE_BRACKET	Символ ']'
TOKEN_AT_SYMBOL	Символ '@'
TOKEN_XML_NAME	Эквивалент типа xs:token схемы XML
TOKEN_UNSIGNED_POSITIVE_INTEGER	Значения в диапазоне 1.. 4294967295
TOKEN_TEXT	Символы 'text'
TOKEN_OPEN_PAREN	Символ '('
TOKEN_CLOSE_PAREN	Символ ')'
TOKEN_AND	Символы 'and'
TOKEN_OR	Символы 'or'
TOKEN_NOT	Символы 'not'
TOKEN_STRING	Эквивалент типа xs:string схемы XML
QUOTE	Символ '"'

Пример:

Данный диалект обеспечивает тот же самый тип синтаксиса выбора как уровень 1, но добавляет фильтрацию, как в следующих примерах, применяемых к примеру документа уровня 1, приведенному выше:

(1) b [@x = «y»] // Выбирают <b>, если у него есть атрибут x = «y»

(2) b [. = «100»] // Выбирают <b>, если содержимое равно 100

(3) c [d = «200»] // Выбирают <c>, если содержимое <d> равно 200

(4) c/d [. = «200»] // Выбирают <d>, если его содержимое равно 200

(5) b [. = «100» and @x = «z»] // Выбирают <b>, если это значение 100 и атрибут @x = «z»

(6) c [d = «200» ord = «300»]//Выбирают все<c> с d=200 или d=300

(7) c[2] [not(. = «400» or@x = «100»)]

(8) //Выберите второй <c>такой, что:

(9) //его значение не 400, и у него нет атрибута x, равного 100

(10) c/d [. = «100» or (@x = «400» and. = «500»)]

(11) //Выберите <d> при условии, что:

(12)//его значение равняется 100 или его значение 500, и у него есть атрибут x, равный 400

В сущности данный диалект позволяет выбирать любой узел, основываясь на выражении фильтра с полным набором операторов отношений, логических операторов и подвыражений в скобках.

Службам, которые поддерживают диалекты фильтрации, основанные на XPath, как описано в настоящем стандарте, рекомендуется поддерживать подмножество XPath, по крайней мере, столь же сильное, как это описано на уровне 2.

Очевидно, служба может предоставить полный XPath 1.0 или любое другое подмножество, которое соответствует или превышает требования, определенные в данном пункте.

В фактических операциях, таких как Enumerate и Subscribe, диалект XPath определен под нормальным URI для полного XPath:

gost.e-ecolog.ru Ǝ/1999/REC-xpath-19991116

Приложение Е
(обязательное)

Диалект фильтра селектора

Диалект фильтра селектора — простой диалект фильтрации, который позволяет перечисление с фильтрацией или подписку без изменения представления.

Селекторы — часть модели адресации по умолчанию, как определено в 5.1. Данный диалект предназначен для реализаций, которые поддерживают модель адресации по умолчанию, так как это дает возможность поддерживать фильтрацию, используя подобный синтаксис, избегая накладных расходов обработки более сложных диалектов.

Данная спецификация определяет следующий URI диалекта фильтра для диалекта селектора:

gost.e-ecolog.ru

Если служба использует модель адресации WS-Management по умолчанию, она может поддерживать данный диалект фильтра для операций подписки и перечисления.

Диалект фильтра селектора может использоваться для определения пары имя-значение в синтаксисе селектора, чтобы отфильтровать результаты запроса перечисления или определить целевые события в запросе подписки. Селекторы действуют как механизм выбора в пространстве класса ресурса, подразумеваемого Resource URI; однако из этого не следует, что значения селектора являются ключами в возвращаемом ресурсе или хотя бы часть его.

Для фильтра в запросе Enumerate используется следующий синтаксис:

```
(1)  <s:Header>
(2)    <wsa:To> Service transport address </wsa:To>
(3)    <wsman:ResourceURI> Resource URI </wsman:ResourceURI>
(4)    ...
(5)  </s:Header>
(6)  <s:Body>
(7)    <wsmen:Enumerate>
(8)      <wsman:Filter
(9)        Dialect=“ gost.e-ecolog.ru                                    >
(10)     <wsman:SelectorSet>
(11)       <wsman:Selector Name=“selector-name”>
(12)         selector-value
(13)       </wsman:Selector> +
(14)     </wsman:SelectorSet>
(15)     </wsman:Filter>
(16)     ...
(17)   </wsmen:Enumerate>
(18) </s:Body>
```

Поскольку синтаксис фильтра не включает информацию о типе ресурса, ResourceURI, определенный в блоке адресации, используется для идентификации типа ресурса. Для определения результата фильтра все селекторы в SelectorSet логически соединяются связкой И.

RE-1 Если поддерживается диалект фильтра селектора, служба должна принимать как имена селектора локальную часть (NCName) квалифицированного имени QNames любого из элементов верхнего уровня, которые представляют ресурс или событие и могут принимать дополнительные имена селекторов. Если служба поддерживает фильтрацию только по подмножеству этих QName и фильтр ссылается на неподержанное QName, то служба должна ответить отказом wsme:CannotProcessFilter (или wsman:CannotProcessFilter для подписки) и должна предоставить в деталях отказа список имен селекторов, которые служба поддерживает для фильтрации.

RE-2 Для каждого имени селектора, определенного в фильтре, результат операции должен содержать только объекты, чье имя элемента равно значению селектора. У элементов, на которые фильтр не ссылается, может быть любое значение.

Возможно, что некоторый ресурс или представления событий включают одноименные элементы, но из различных пространств имен XML. В этом случае служба может выбрать для сравнения любой из элементов, где тип соответствует предоставленному селектору. Клиенты могут быть написаны таким образом, чтобы учитывать такое поведение и осуществлять дополнительную обработку, необходимую для определения набора желаемых случаев.

RE-3 Если представление ресурса или события будут включать два или больше элемента с QNames с идентичной локальной частью, но чьи имена пространства имен отличаются, и все следующие условия выполняются, то служба не должна отклонять запрос и должна обработать фильтр таким образом, что это соответствует точно одному из элементов, для которых фильтрация поддержана, используя алгоритм выбора по своему усмотрению:

- фильтр селектора содержит элемент wsman:Selector, атрибут name которого соответствует местной части каждого из этих элементов;
- по крайней мере, у одного из соответствующих элементов есть тип и пространство значений, совместимое с предоставленными типом селектора и значением;
- служба поддерживает фильтрацию, по крайней мере, на одном из соответствующих элементов в соответствии с RE-1.

RE-4 Если представление ресурса или события включает элементы типа массив, и фильтр содержит элемент wsman:Selector, атрибут name которого соответствует локальной части QName этих элементов, и служба поддерживает фильтрацию в соответствии с RE-1, то служба должна обработать фильтр таким образом, что результаты включают все представления, для которых по крайней мере у одного элемента массива есть значение, равное значению селектора.

Обработка элемента SelectorSet, когда тот используется в качестве фильтра, следует тем же правилам обработки, как в EPR (как описано в 5.4.2), относительно двойных имен селектора, несоответствий типов, неожиданных селекторов, ограничений размера и т. д.

RE-5 Если выражение фильтра содержит SelectorSet, который не соответствует правилам 5.4.2, службе следует сообщить об ошибке wsme:CannotProcessFilter (или wsman:CannotProcessFilter для подписки) с соответствующим кодом детализации.

Приложение F
(справочное)

XML-схема для идентификации

Нормативная копия XML-схемы ответного сообщения идентификации может быть получена по следующему адресу:

gost.e-ecolog.ru

Следующая ненормативная копия XML-схемы предоставлена для удобства:

(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!--
(3) Notice
(4) DSP8012
(5) Document: WS-Management Identify XML Schema
(6) Version: 1.0.1
(7) Status: Final
(8) Date: 02/27/2009
(9) Author: DMTF WS-Management Work Group Email:gost.e-ecolog.ru
(10) Description: XML Schema for WS-Management Identify Operation.
(11)
(12) Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved. DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted. Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations. For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit

gost.e-ecolog.ru

(13)
(14) -->

```
(15) <xs:schema
(16)  targetNamespace="gost.e-ecolog.ru                                    ity.xsd"
(17)
xmlns:wsmid="gost.e-ecolog.ru
(18)     xmlns:xs="gost.e-ecolog.ru
(19)  elementFormDefault="qualified" version="1.0.1">
(20)  <xs:complexType name="IdentifyType">
(21)   <xs:sequence>
(22)    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
(23)     processContents="lax" />
(24)   </xs:sequence>
(25)   <xs:anyAttribute namespace="##other" processContents="lax" />
(26)  </xs:complexType>
(27)  <xs:element name="Identify" type="wsmid:IdentifyType" />
(28)
(29)  <xs:simpleType name="restrictedProtocolVersionType">
(30)
(31)   <xs:restriction base="xs:anyURI">
(32)    <xs:enumeration
(33)
5956 value="gost.e-ecolog.ru                                          usDisclosure" />
(34)   </xs:restriction>
(35)  </xs:simpleType>
(36)
(37)  <xs:simpleType name="ProtocolVersionType">
(38)   <xs:union memberTypes="wsmid:restrictedProtocolVersionType xs:anyURI" />
(39)
(40)  </xs:simpleType>
(41)  <xs:element name="ProtocolVersion" type="wsmid:ProtocolVersionType" />
(42)  <xs:element name="ProductVendor" type="xs:string" />
(43)  <xs:element name="ProductVersion" type="xs:string" />
(44)  <xs:element name="InitiativeName" type="xs:string" />
(45)  <xs:element name="InitiativeVersion" type="wsmid:VERSION_VALUE"/>
(46)  <xs:element name="SecurityProfileName" type="xs:anyURI" />
(47)  <xs:complexType name="SecurityProfilesType">
(48)   <xs:sequence>
(49)
(50)    <xs:element ref="wsmid:SecurityProfileName" minOccurs="0"
(51)     maxOccurs="unbounded" />
(52)   </xs:sequence>
(53)  </xs:complexType>
(54)  <xs:element name="SecurityProfiles" type="wsmid:SecurityProfilesType" />
(55)  <xs:element name="AddressingVersionURI" type="xs:anyURI" />
(56)  <xs:element name="IntiativeSupport">
(57)   <xs:complexType>
(58)    <xs:sequence>
(59)     <xs:element ref="wsmid:InitiativeName" minOccurs="0" maxOccurs="1" />
(60)
(61)     <xs:element ref="wsmid:InitiativeVersion" minOccurs="0" maxOccurs="1"/>
```

```
(62)      </xs:sequence>
(63)    </xs:complexType>
(64)  </xs:element>
(65)
(66)  <xs:complexType name="IdentifyResponseType">
(67)    <xs:sequence>
(68)      <xs:element ref="wsmid:ProtocolVersion" maxOccurs="unbounded" />
(69)      <xs:element ref="wsmid:ProductVendor" minOccurs="0" />
(70)      <xs:element ref="wsmid:ProductVersion" minOccurs="0" />
(71)
(72)      <xs:element ref="wsmid:IntiativeSupport" minOccurs="0" maxOccurs="unbounded"/>
(73)      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" />
(74)      <xs:element ref="wsmid:SecurityProfiles" minOccurs="0"
(75)        maxOccurs="1" />
(76)      <xs:element ref="wsmid:AddressingVersionURI" minOccurs="0"
(77)        maxOccurs="unbounded" />
(78)    </xs:sequence>
(79)    <xs:anyAttribute namespace="##other" processContents="lax" />
(80)  </xs:complexType>
(81)
(82)  <xs:element name="IdentifyResponse" type="wsmid:IdentifyResponseType" />
(83)
(84)  <xs:simpleType name="VERSION_VALUE">
(85)
(86)    <xs:annotation>
(87)      <xs:documentation>Version values must be in form of M.N.U (Major, Minor, Update)</xs:documentation>
(88)    </xs:annotation>
(89)    <xs:restriction base="xs:string">
(90)      <xs:pattern value="\d*.\d*.\d*" />
(91)    </xs:restriction>
(92)  </xs:simpleType>
(93)
(94) </xs:schema>
```

Приложение G
(справочное)

XML-схема и WSDL для операций доступа к ресурсу

Нормативная копия XML-схем (XML-схема 1, XML-схема 2) для операций доступа к ресурсу может быть получена по следующему адресу:

gost.e-ecolog.ru

Следующая ненормативная копия XML-схемы предоставлена для удобства:

```
(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!--
(3) DMTF — Distributed Management Task Force, Inc. — gost.e-ecolog.ru
(4)
(5) Document number: DSP8031
(6) Date: 2010-02-19
(7) Version: 1.0.0
(8) Document status: DMTF Standard
(9)
(10) Title: WS-Management Resource Access Operations XML Schema
(11)
(12) Document type: Specification (W3C XML Schema)
(13) Document language: E
(14)
(15) Abstract: XML Schema for WS-Management Resource Access Operations.
(16)
(17) Contact group: DMTF WS-Management Work Group, gost.e-ecolog.ru
(18)
(19) Copyright (C) 2008—2010 Distributed Management Task Force, Inc. (DMTF).
(20) All rights reserved. DMTF is a not-for-profit association of industry
(21) members dedicated to promoting enterprise and systems management and
(22) interoperability. Members and non-members may reproduce DMTF
(23) specifications and documents, provided that correct attribution is
(24) given. As DMTF specifications may be revised from time to time,
(25) the particular version and release date should always be noted.
(26) Implementation of certain elements of this standard or proposed
(27) standard may be subject to third party patent rights, including
(28) provisional patent rights (herein "patent rights"). DMTF makes
(29) no representations to users of the standard as to the existence
(30) of such rights, and is not responsible to recognize, disclose,
(31) or identify any or all such third party patent right, owners or
(32) claimants, nor for any incomplete or inaccurate identification or
(33) disclosure of such rights, owners or claimants. DMTF shall have no
(34) liability to any party, in any manner or circumstance, under any legal
(35) theory whatsoever, for failure to recognize, disclose, or identify any
(36) such third party patent rights, or for such party's reliance on the
(37) standard or incorporation thereof in its product, protocols or testing
(38) procedures. DMTF shall have no liability to any party implementing
(39) such standard, whether such implementation is foreseeable or not, nor
(40) to any patent owner or claimant, and shall have no liability or
```

```
(41) responsibility for costs or losses incurred if a standard is withdrawn
(42) or modified after publication, and shall be indemnified and held
(43) harmless by any party implementing the standard from any and all claims
(44) of infringement by a patent owner for such implementations.  For
(45) information about patents held by third-parties which have notified the
(46) DMTF that, in their opinion, such patent may relate to or impact
(47) implementations of DMTF standards, visit
(48) gost.e-ecolog.ru
(49)
(50) Change log:
(51) 1.0.0 — 2009-11-01 — Work in Progress release
(52) 1.0.0 — 2010-02-19 — DMTF Standard release
(53)
(54)  -->
(55) <xs:schema
(56)  targetNamespace="gost.e-ecolog.ru
(57)  xmlns:tns="gost.e-ecolog.ru
(58)  xmlns:xs="gost.e-ecolog.ru
(59)  xmlns:wsa04="gost.e-ecolog.ru
(60)  xmlns:wsa10="gost.e-ecolog.ru
(61)  elementFormDefault="qualified"
(62)  blockDefault="#all" >
(63)
(64)  <xs:import
(65)   namespace="gost.e-ecolog.ru
(66)   schemaLocation="gost.e-ecolog.ru                                    />
(67)  <xs:import
(68)   namespace="gost.e-ecolog.ru
(69)   schemaLocation="gost.e-ecolog.ru                                  />
(70)
(71) <!--
(72)  The type of the AnyEPRType is effectively
(73)  the union of wsa04:EndpointReferenceType and
(74)  wsa10:EndpointReferenceType. Unfortunately, xs:union only
(75)  works for simple types. As a result, we have to define
(76)  the element in an unvalidated way to accommodate either
(77)  addressing type.
(78)  -->
(79)
(80)   <xs:complexType name="AnyEPRType">
(81)    <xs:sequence>
(82)     <xs:any minOccurs='1' maxOccurs='unbounded' processContents='skip'
(83)      namespace='##other' />
(84)    </xs:sequence>
(85)  </xs:complexType>
(86)
(87)  <xs:element name="ResourceCreated" type="tns:AnyEPRType"/>
(88)
(89)  <!-- The following GED is defined for convenience. This GED
```

```
(90)        may be used in cases where a resource-specific GED is
(91)        not available. -->
(92)    <xs:element name="TransferElement">
(93)      <xs:complexType>
(94)        <xs:sequence>
(95)          <xs:any minOccurs='1' maxOccurs='unbounded'
(96)            processContents='skip' namespace='##other'/>
(97)        </xs:sequence>
(98)      </xs:complexType>
(99)    </xs:element>
(100)
(101)  </xs:schema>
```

Нормативная копия описания WSDL для операций доступа к ресурсу может быть получена по следующему адресу:

gost.e-ecolog.ru

Следующая ненормативная копия описания WSDL предоставлена для удобства:

```
(1)  <?xml version="1.0" encoding="UTF-8"?>
(2)  <!--
(3)  DMTF — Distributed Management Task Force, Inc. —
(4)
(5)  Document number: DSP8035
(6)  Date: 2010-02-19
(7)  Version: 1.0.0
(8)  Document status: DMTF Standard
(9)
(10) Title: WS-Management Resource Access Operations WSDL
(11)
(12) Document type: Specification (W3C WSDL Document)
(13) Document language: E
(14)
(15) Abstract: WSDL for WS-Management Resource Access Operations.
(16)
(17) Contact group: DMTF WS-Management Work Group,
(18)
(19) Copyright (C) 2008—2010 Distributed Management Task Force, Inc. (DMTF).
(20) All rights reserved.  DMTF is a not-for-profit association of industry
(21) members dedicated to promoting enterprise and systems management and
(22) interoperability.  Members and non-members may reproduce DMTF
(23) specifications and documents, provided that correct attribution is
(24) given.  As DMTF specifications may be revised from time to time,
(25) the particular version and release date should always be noted.
(26) Implementation of certain elements of this standard or proposed
(27) standard may be subject to third party patent rights, including
(28) provisional patent rights (herein "patent rights").  DMTF makes
(29) no representations to users of the standard as to the existence
(30) of such rights, and is not responsible to recognize, disclose,
(31) or identify any or all such third party patent right, owners or
(32) claimants, nor for any incomplete or inaccurate identification or
(33) disclosure of such rights, owners or claimants.  DMTF shall have no
```

```
(34) liability to any party, in any manner or circumstance, under any legal
(35) theory whatsoever, for failure to recognize, disclose, or identify any
(36) such third party patent rights, or for such party's reliance on the
(37) standard or incorporation thereof in its product, protocols or testing
(38) procedures.  DMTF shall have no liability to any party implementing
(39) such standard, whether such implementation is foreseeable or not, nor
(40) to any patent owner or claimant, and shall have no liability or
(41) responsibility for costs or losses incurred if a standard is withdrawn
(42) or modified after publication, and shall be indemnified and held
(43) harmless by any party implementing the standard from any and all claims
(44) of infringement by a patent owner for such implementations.  For
(45) information about patents held by third-parties which have notified the
(46) DMTF that, in their opinion, such patent may relate to or impact
(47) implementations of DMTF standards, visit
(48) gost.e-ecolog.ru
(49)
(50) Change log:
(51) 1.0.0 — 2009-11-01 — Work in Progress release
(52) 1.0.0 — 2010-02-19 — DMTF Standard release
(53)
(54)  -->
(55) <wsdl:definitions
(56)    targetNamespace=" gost.e-ecolog.ru
(57)    xmlns:tns=" gost.e-ecolog.ru
(58)    xmlns:wsa=" gost.e-ecolog.ru
(59)    xmlns:wsam=" gost.e-ecolog.ru
(60)    xmlns:wsdl=" gost.e-ecolog.ru
(61)    xmlns:xs=" gost.e-ecolog.ru                                    >
(62)
(63)  <wsdl:types>
(64)    <xs:schema>
(65)      <xs:import
(66)        namespace=" gost.e-ecolog.ru
(67)
schemaLocation=" gost.e-ecolog.ru
(68)        />
(69)    </xs:schema>
(70)  </wsdl:types>
(71)
(72)  <!–
(73)  In some of the messages defined below a "resource-specific-GED"
(74)  is expected to be inserted before the WSDL is processed by any tooling.
(75)  Thus the WSDL as presented is not usable until after this substitution
(76)  is done.
(77)  -->
(78)
(79)  <wsdl:message name="EmptyMessage"/>
(80)  <wsdl:message name="CreateRequestMessage">
(81)    <wsdl:part name="Body" element="resource-specific-GED"/>
```

```
(82)  </wsdl:message>
(83)  <wsdl:message name="CreateResponseMessage">
(84)    <wsdl:part name="Body" element="tns:ResourceCreated"/>
(85)  </wsdl:message>
(86)  <wsdl:message name="GetResponseMessage">
(87)    <wsdl:part name="Body" element="resource-specific-GED"/>
(88)  </wsdl:message>
(89)  <wsdl:message name="PutRequestMessage">
(90)    <wsdl:part name="Body" element="resource-specific-GED"/>
(91)  </wsdl:message>
(92)  <wsdl:message name="PutResponseMessage">
(93)    <!-- Note this 'part' may be omitted -->
(94)    <wsdl:part name="Body" element="resource-specific-GED"/>
(95)  </wsdl:message>
(96)
(97)  <wsdl:portType name="Resource">
(98)    <wsdl:documentation>
(99)      This port type defines a resource that may be read,
(100)       written, and deleted.
(101)     </wsdl:documentation>
(102)     <wsdl:operation name="Get">
(103)       <wsdl:input
(104)         message="tns:EmptyMessage"
(105)         wsa:Action="gost.e-ecolog.ru
(106)         wsam:Action="gost.e-ecolog.ru                    />
(107)       <wsdl:output
(108)         message="tns:GetResponseMessage"
(109)
wsa:Action="gost.e-ecolog.ru
(110)
wsam:Action="gost.e-ecolog.ru                                  />
(111)     </wsdl:operation>
(112)     <wsdl:operation name="Put">
(113)       <wsdl:input
(114)         message="tns:PutRequestMessage"
(115)         wsa:Action="gost.e-ecolog.ru
```

© ISO/IEC 2013 — All rights reserved ISO/IEC 17963:2013(E)

185

Web Services for Management (WS-Management) Specification DSP0226

186 DMTF Standard Version 1.1.1

```
(116)         wsam:Action="gost.e-ecolog.ru                    />
(117)       <wsdl:output
(118)         message="tns:PutResponseMessage"
(119)
wsa:Action="gost.e-ecolog.ru
(120)
wsam:Action="gost.e-ecolog.ru                                  >
(121)     </wsdl:operation>
(122)     <wsdl:operation name="Delete">
```

```
(123)       <wsdl:input
(124)         message="tns:EmptyMessage"
(125)
wsa:Action="gost.e-ecolog.ru
(126)
wsam:Action="gost.e-ecolog.ru                                    />
(127)       <wsdl:output
(128)         message="tns:EmptyMessage"
(129)
wsa:Action="gost.e-ecolog.ru
(130)
wsam:Action="gost.e-ecolog.ru
(131)       />
(132)     </wsdl:operation>
(133)   </wsdl:portType>
(134)
(135)   <wsdl:portType name="ResourceFactory">
(136)     <wsdl:documentation>
(137)       This port type defines a Web service that can create new
(138)       resources.
(139)     </wsdl:documentation>
(140)     <wsdl:operation name="Create">
(141)       <wsdl:input
(142)         message="tns:CreateRequestMessage"
(143)
wsa:Action="gost.e-ecolog.ru
(144)
wsam:Action="gost.e-ecolog.ru                                    />
(145)       <wsdl:output
(146)         message="tns:CreateResponseMessage"
(147)
wsa:Action="gost.e-ecolog.ru
(148)
wsam:Action="gost.e-ecolog.ru
(149)       />
(150)     </wsdl:operation>
(151)   </wsdl:portType>
(152) </wsdl:definitions>
```

Приложение Н
(справочное)

XML-схема и WSDL операций перечисления

Нормативная копия XML-схем для операций перечисления может быть получена по следующему:
gost.e-ecolog.ru
Следующая ненормативная копия XML-схемы предоставлена для удобства:

```
(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!--
(3) DMTF — Distributed Management Task Force, Inc. — gost.e-ecolog.ru
(4)
(5) Document number: DSP8033
(6) Date: 2010-02-19
(7) Version: 1.0.0
(8) Document status: DMTF Standard
(9)
(10) Title: WS-Management Enumeration Operations XML Schema
(11)
(12) Document type: Specification (W3C XML Schema)
(13) Document language: E
(14)
(15) Abstract: XML Schema for WS-Management Enumeration Operations.
(16)
(17) Contact group: DMTF WS-Management Work Group, gost.e-ecolog.ru
(18)
(19) Copyright (C) 2008—2010 Distributed Management Task Force, Inc. (DMTF).
(20) All rights reserved.  DMTF is a not-for-profit association of industry
(21) members dedicated to promoting enterprise and systems management and
(22) interoperability.  Members and non-members may reproduce DMTF
(23) specifications and documents, provided that correct attribution is
(24) given.  As DMTF specifications may be revised from time to time,
(25) the particular version and release date should always be noted.
(26) Implementation of certain elements of this standard or proposed
(27) standard may be subject to third party patent rights, including
(28) provisional patent rights (herein "patent rights").  DMTF makes
(29) no representations to users of the standard as to the existence of
(30) such rights, and is not responsible to recognize, disclose,
(31) or identify any or all such third party patent right, owners or
(32) claimants, nor for any incomplete or inaccurate identification or
(33) disclosure of such rights, owners or claimants.  DMTF shall have no
(34) liability to any party, in any manner or circumstance, under any legal
(35) theory whatsoever, for failure to recognize, disclose, or identify any
(36) such third party patent rights, or for such party's reliance on the
(37) standard or incorporation thereof in its product, protocols or testing
(38) procedures.  DMTF shall have no liability to any party implementing
(39) such standard, whether such implementation is foreseeable or not, nor
(40) to any patent owner or claimant, and shall have no liability or
(41) responsibility for costs or losses incurred if a standard is withdrawn
```

```
(42) or modified after publication, and shall be indemnified and held
(43) harmless by any party implementing the standard from any and all claims
(44) of infringement by a patent owner for such implementations.  For
(45) information about patents held by third-parties which have notified the
(46) DMTF that, in their opinion, such patent may relate to or impact
(47) implementations of DMTF standards, visit
(48) gost.e-ecolog.ru
(49)
(50) Change log:
(51) 1.0.0 — 2009-11-01 — Work in Progress release
(52) 1.0.0 — 2010-02-19 — DMTF Standard release
(53)
(54)  -->
(55) <xs:schema
(56)    targetNamespace="gost.e-ecolog.ru
(57)    xmlns:tns="gost.e-ecolog.ru
(58)    xmlns:wsa="gost.e-ecolog.ru
(59)    xmlns:xs="gost.e-ecolog.ru
(60)    elementFormDefault="qualified"
(61)    blockDefault="#all">
(62)
(63)  <xs:import
(64)    namespace="gost.e-ecolog.ru
(65)    schemaLocation="gost.e-ecolog.ru                    />
(66)  <xs:import
(67)    namespace="gost.e-ecolog.ru
(68)    schemaLocation="gost.e-ecolog.ru                                    />
(69)  <xs:import
(70)     namespace="gost.e-ecolog.ru
(71)     schemaLocation="gost.e-ecolog.ru                                  />
(72)
(73)  <!-- Types and global elements -->
(74)  <xs:complexType name="FilterType" mixed="true">
(75)    <xs:sequence>
(76)      <xs:any namespace="##other" processContents="lax"
(77)            minOccurs="0" maxOccurs="unbounded" />
(78)    </xs:sequence>
(79)    <xs:attribute name="Dialect" type="xs:anyURI" />
(80)    <xs:anyAttribute namespace="##other" processContents="lax" />
(81)  </xs:complexType>
(82)
(83)  <xs:simpleType name="PositiveDurationType">
(84)    <xs:restriction base="xs:duration">
(85)      <xs:minExclusive value="P0Y0M0DT0H0M0S" />
(86)    </xs:restriction>
(87)  </xs:simpleType>
(88)
(89)  <xs:simpleType name="NonNegativeDurationType">
(90)    <xs:restriction base="xs:duration">
```

```
(91)      <xs:minInclusive value="P0Y0M0DT0H0M0S" />
(92)    </xs:restriction>
(93)  </xs:simpleType>
(94)
(95)  <xs:simpleType name="ExpirationType">
(96)    <xs:union memberTypes="xs:dateTime tns:NonNegativeDurationType" />
(97)  </xs:simpleType>
(98)
(99)  <xs:complexType name="EnumerationContextType">
(100)     <xs:complexContent mixed="true">
(101)       <xs:restriction base="xs:anyType">
(102)         <xs:sequence>
(103)           <xs:any namespace="##other" processContents="lax"
(104)                 minOccurs="0" maxOccurs="unbounded" />
(105)         </xs:sequence>
(106)         <xs:anyAttribute namespace="##other" processContents="lax" />
(107)       </xs:restriction>
(108)     </xs:complexContent>
(109)   </xs:complexType>
(110)
(111)   <xs:complexType name="ItemListType">
(112)     <xs:sequence maxOccurs="unbounded">
(113)       <xs:any namespace="##other" processContents="lax"
(114)             minOccurs="0" maxOccurs="unbounded" />
(115)     </xs:sequence>
(116)   </xs:complexType>
(117)
(118)   <xs:complexType name="LanguageSpecificStringType">
(119)     <xs:simpleContent>
(120)       <xs:extension base="xs:string">
(121)         <xs:attribute ref="xml:lang" />
(122)         <xs:anyAttribute namespace="##other" processContents="lax" />
(123)       </xs:extension>
(124)     </xs:simpleContent>
(125)   </xs:complexType>
(126)
(127)   <!--
(128)     The type of the AnyEPRType is effectively
(129)     the union of wsa04:EndpointReferenceType and
(130)     wsa10:EndpointReferenceType. Unfortunately, xs:union only
(131)     works for simple types. As a result, we have to define
(132)     the element in an unvalidated way to accommodate either
(133)     addressing type.
(134)     -->
(135)
(136)   <xs:complexType name="AnyEPRType">
(137)     <xs:sequence>
(138)       <xs:any minOccurs='1' maxOccurs='unbounded' processContents='skip'
(139)             namespace='##other' />
```

```
(140)     </xs:sequence>
(141)   </xs:complexType>
(142)
(143)   <!-- Enumerate request -->
(144)   <xs:element name="Enumerate">
(145)     <xs:complexType>
(146)       <xs:sequence>
(147)         <xs:element name="EndTo" type="tns:AnyEPRType"
(148)                 minOccurs="0" />
(149)         <xs:element name="Expires" type="tns:ExpirationType"
(150)                 minOccurs="0" />
(151)         <xs:element name="Filter" type="tns:FilterType"
(152)                 minOccurs="0" />
(153)         <xs:any namespace="##other" processContents="lax"
(154)               minOccurs="0" maxOccurs="unbounded" />
(155)       </xs:sequence>
(156)       <xs:anyAttribute namespace="##other" processContents="lax" />
(157)     </xs:complexType>
(158)   </xs:element>
(159)
(160)   <!-- Used for a fault response -->
(161)   <xs:element name="SupportedDialect" type="xs:anyURI" />
(162)
(163)   <!-- Enumerate response -->
(164)   <xs:element name="EnumerateResponse">
(165)     <xs:complexType>
(166)       <xs:sequence>
(167)         <xs:element name="Expires" type="tns:ExpirationType"
(168)                 minOccurs="0" />
(169)         <xs:element name="EnumerationContext"
(170)                 type="tns:EnumerationContextType" />
(171)         <xs:any namespace="##other" processContents="lax"
(172)               minOccurs="0" maxOccurs="unbounded" />
(173)       </xs:sequence>
(174)       <xs:anyAttribute namespace="##other" processContents="lax" />
(175)     </xs:complexType>
(176)   </xs:element>
(177)
(178)   <!-- Pull request -->
(179)   <xs:element name="Pull">
(180)     <xs:complexType>
(181)       <xs:sequence>
(182)         <xs:element name="EnumerationContext"
(183)                 type="tns:EnumerationContextType" />
(184)         <xs:element name="MaxTime" type="tns:PositiveDurationType"
(185)                 minOccurs="0" />
(186)         <xs:element name="MaxElements" type="xs:positiveInteger"
(187)                 minOccurs="0" />
(188)         <xs:element name="MaxCharacters" type="xs:positiveInteger"
```

```
(189)                minOccurs="0" />
(190)       <xs:any namespace="##other" processContents="lax"
(191)             minOccurs="0" maxOccurs="unbounded" />
(192)     </xs:sequence>
(193)     <xs:anyAttribute namespace="##other" processContents="lax" />
(194)   </xs:complexType>
(195)  </xs:element>
(196)
(197)  <!-- Pull response -->
(198)  <xs:element name="PullResponse">
(199)   <xs:complexType>
(200)     <xs:sequence>
(201)       <xs:element name="EnumerationContext"
(202)                type="tns:EnumerationContextType"
(203)                minOccurs="0" />
(204)       <xs:element name="Items" type="tns:ItemListType"
(205)                minOccurs="0" />
(206)       <xs:element name="EndOfSequence" minOccurs="0" />
(207)     </xs:sequence>
(208)     <xs:anyAttribute namespace="##other" processContents="lax" />
(209)   </xs:complexType>
(210)  </xs:element>
(211)
(212)  <!-- Renew request -->
(213)  <xs:element name="Renew">
(214)   <xs:complexType>
(215)     <xs:sequence>
(216)       <xs:element name="EnumerationContext"
(217)                type="tns:EnumerationContextType" />
(218)       <xs:element name="Expires" type="tns:ExpirationType"
(219)                minOccurs="0" />
(220)       <xs:any namespace="##other" processContents="lax"
(221)             minOccurs="0" maxOccurs="unbounded" />
(222)     </xs:sequence>
(223)     <xs:anyAttribute namespace="##other" processContents="lax" />
(224)   </xs:complexType>
(225)  </xs:element>
(226)
(227)  <!-- Renew response -->
(228)  <xs:element name="RenewResponse">
(229)   <xs:complexType>
(230)     <xs:sequence>
(231)       <xs:element name="Expires" type="tns:ExpirationType"
(232)                minOccurs="0" />
(233)       <xs:element name="EnumerationContext"
(234)                type="tns:EnumerationContextType"
(235)                minOccurs="0" />
(236)       <xs:any namespace="##other" processContents="lax"
(237)             minOccurs="0" maxOccurs="unbounded" />
```

```
(238)      </xs:sequence>
(239)      <xs:anyAttribute namespace="##other" processContents="lax" />
(240)    </xs:complexType>
(241)  </xs:element>
(242)
(243)  <!-- GetStatus request -->
(244)  <xs:element name="GetStatus">
(245)    <xs:complexType>
(246)      <xs:sequence>
(247)        <xs:element name="EnumerationContext"
(248)                type="tns:EnumerationContextType" />
(249)        <xs:any namespace="##other" processContents="lax"
(250)              minOccurs="0" maxOccurs="unbounded" />
(251)      </xs:sequence>
(252)      <xs:anyAttribute namespace="##other" processContents="lax" />
(253)    </xs:complexType>
(254)  </xs:element>
(255)
(256)  <!-- GetStatus response -->
(257)  <xs:element name="GetStatusResponse">
(258)    <xs:complexType>
(259)      <xs:sequence>
(260)        <xs:element name="Expires" type="tns:ExpirationType"
(261)                minOccurs="0" />
(262)        <xs:any namespace="##other" processContents="lax"
(263)              minOccurs="0" maxOccurs="unbounded" />
(264)      </xs:sequence>
(265)      <xs:anyAttribute namespace="##other" processContents="lax" />
(266)    </xs:complexType>
(267)  </xs:element>
(268)
(269)  <!-- Release request -->
(270)  <xs:element name="Release">
(271)    <xs:complexType>
(272)      <xs:sequence>
(273)        <xs:element name="EnumerationContext"
(274)                type="tns:EnumerationContextType" />
(275)      </xs:sequence>
(276)      <xs:anyAttribute namespace="##other" processContents="lax" />
(277)    </xs:complexType>
(278)  </xs:element>
(279)
(280)  <!-- Release response has an empty body -->
(281)
(282)  <!-- EnumerationEnd message -->
(283)  <xs:element name="EnumerationEnd">
(284)    <xs:complexType>
(285)    <xs:sequence>
(286)      <xs:element name="EnumerationContext"
```

```
(287)                type="tns:EnumerationContextType" />
(288)      <xs:element name="Code" type="tns:OpenEnumerationEndCodeType" />
(289)      <xs:element name="Reason" type="tns:LanguageSpecificStringType"
(290)                minOccurs="0" maxOccurs="unbounded" />
(291)      <xs:any namespace="##other" processContents="lax"
(292)            minOccurs="0" maxOccurs="unbounded" />
(293)    </xs:sequence>
(294)    <xs:anyAttribute namespace="##other" processContents="lax" />
(295)    </xs:complexType>
(296)  </xs:element>
(297)
(298)  <xs:simpleType name="EnumerationEndCodeType">
(299)    <xs:restriction base="xs:anyURI">
(300)    <xs:enumeration
value="gost.e-ecolog.ru                                                    />
(301)    <xs:enumeration
value="gost.e-ecolog.ru                                                  />
(302)    </xs:restriction>
(303)  </xs:simpleType>
(304)
(305)  <xs:simpleType name="OpenEnumerationEndCodeType">
(306)    <xs:union memberTypes="tns:EnumerationEndCodeType xs:anyURI" />
(307)  </xs:simpleType>
(308) </xs:schema>
```

Нормативная копия описания WSDL для операций перечисления может быть получена по адресу: gost.e-ecolog.ru

Следующая ненормативная копия описания WSDL предоставлена для удобства

```
(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!--
(3) DMTF — Distributed Management Task Force, Inc. — gost.e-ecolog.ru
(4)
(5) Document number: DSP8037
(6) Date: 2010-02-19
(7) Version: 1.0.0
(8) Document status: DMTF Standard
(9)
(10) Title: WS-Management Enumeration Operations WSDL
(11)
(12) Document type: Specification (W3C WSDL Document)
(13) Document language: E
(14)
(15) Abstract: WSDL for WS-Management Enumeration Operations.
(16)
(17) Contact group: DMTF WS-Management Work Group, gost.e-ecolog.ru
(18)
(19) Copyright (C) 2008—2010 Distributed Management Task Force, Inc. (DMTF).
(20) All rights reserved.  DMTF is a not-for-profit association of industry
(21) members dedicated to promoting enterprise and systems management and
(22) interoperability.  Members and non-members may reproduce DMTF
```

```
(23) specifications and documents, provided that correct attribution is
(24) given.  As DMTF specifications may be revised from time to time,
(25) the particular version and release date should always be noted.
(26) Implementation of certain elements of this standard or proposed
(27) standard may be subject to third party patent rights, including
(28) provisional patent rights (herein “patent rights”).  DMTF makes
(29) no representations to users of the standard as to the existence of
(30) such rights, and is not responsible to recognize, disclose,
(31) or identify any or all such third party patent right, owners or
(32) claimants, nor for any incomplete or inaccurate identification or
(33) disclosure of such rights, owners or claimants.  DMTF shall have no
(34) liability to any party, in any manner or circumstance, under any legal
(35) theory whatsoever, for failure to recognize, disclose, or identify any
(36) such third party patent rights, or for such party’s reliance on the
(37) standard or incorporation thereof in its product, protocols or testing
(38) procedures.  DMTF shall have no liability to any party implementing
(39) such standard, whether such implementation is foreseeable or not, nor
(40) to any patent owner or claimant, and shall have no liability or
(41) responsibility for costs or losses incurred if a standard is withdrawn
(42) or modified after publication, and shall be indemnified and held
(43) harmless by any party implementing the standard from any and all claims
(44) of infringement by a patent owner for such implementations.  For
(45) information about patents held by third-parties which have notified the
(46) DMTF that, in their opinion, such patent may relate to or impact
(47) implementations of DMTF standards, visit
(48) gost.e-ecolog.ru
(49)
(50) Change log:
(51) 1.0.0 — 2009-11-01 — Work in Progress release
(52) 1.0.0 — 2010-02-19 — DMTF Standard release
(53)
(54)  -->
(55) <wsdl:definitions
(56)    targetNamespace=“ gost.e-ecolog.ru
(57)    xmlns:wsa=“ gost.e-ecolog.ru
(58)    xmlns:wsam=“ gost.e-ecolog.ru
(59)    xmlns:wsdl=“ gost.e-ecolog.ru
(60)    xmlns:wsmen=“ gost.e-ecolog.ru
(61)    xmlns:xs=“ gost.e-ecolog.ru                                    >
(62)
(63)  <wsdl:types>
(64)    <xs:schema>
(65)      <xs:import
(66)        namespace=“gost.e-ecolog.ru
(67)     schemaLocation=“gost.e-ecolog.ru
(68)        />
(69)    </xs:schema>
(70)  </wsdl:types>
(71)
```

```
(72)  <wsdl:message name="EnumerateMessage">
(73)    <wsdl:part name="Body" element="wsmen:Enumerate" />
(74)  </wsdl:message>
(75)  <wsdl:message name="EnumerateResponseMessage">
(76)    <wsdl:part name="Body" element="wsmen:EnumerateResponse" />
(77)  </wsdl:message>
(78)  <wsdl:message name="PullMessage">
(79)    <wsdl:part name="Body" element="wsmen:Pull" />
(80)  </wsdl:message>
(81)  <wsdl:message name="PullResponseMessage">
(82)    <wsdl:part name="Body" element="wsmen:PullResponse" />
(83)  </wsdl:message>
(84)  <wsdl:message name="RenewMessage" >
(85)    <wsdl:part name="Body" element="wsmen:Renew" />
(86)  </wsdl:message>
(87)  <wsdl:message name="RenewResponseMessage" >
(88)    <wsdl:part name="Body" element="wsmen:RenewResponse" />
(89)  </wsdl:message>
(90)  <wsdl:message name="GetStatusMessage" >
(91)    <wsdl:part name="Body" element="wsmen:GetStatus" />
(92)  </wsdl:message>
(93)  <wsdl:message name="GetStatusResponseMessage" >
(94)    <wsdl:part name="Body" element="wsmen:GetStatusResponse" />
(95)  </wsdl:message>
(96)  <wsdl:message name="ReleaseMessage">
(97)    <wsdl:part name="Body" element="wsmen:Release" />
(98)  </wsdl:message>
(99)  <wsdl:message name="ReleaseResponseMessage" />
(100)   <wsdl:message name="EnumerationEndMessage" >
(101)     <wsdl:part name="Body" element="wsmen:EnumerationEnd" />
(102)   </wsdl:message>
(103)
(104)   <wsdl:portType name="DataSource">
(105)     <wsdl:operation name="EnumerateOp">
(106)       <wsdl:input
(107)         message="wsmen:EnumerateMessage"
(108)
wsa:Action="gost.e-ecolog.ru
(109)
wsam:Action="gost.e-ecolog.ru
(110)         />
(111)       <wsdl:output
(112)         message="wsmen:EnumerateResponseMessage"
(113)
wsa:Action="gost.e-ecolog.ru
(114)
wsam:Action="gost.e-ecolog.ru
(115)         />
(116)     </wsdl:operation>
```

```
(117)     <wsdl:operation name="PullOp">
(118)       <wsdl:input
(119)         message="wsmen:PullMessage"
(120)
wsa:Action="gost.e-ecolog.ru
(121)
wsam:Action="gost.e-ecolog.ru
(122)         />
(123)       <wsdl:output
(124)         message="wsmen:PullResponseMessage"
(125)
wsa:Action="gost.e-ecolog.ru
(126)
wsam:Action="gost.e-ecolog.ru
(127)         />
(128)     </wsdl:operation>
(129)     <wsdl:operation name="RenewOp" >
(130)       <wsdl:input
(131)         message="wsmen:RenewMessage"
(132)
wsa:Action="gost.e-ecolog.ru
(133)
wsam:Action="gost.e-ecolog.ru
(134)         />
(135)       <wsdl:output
(136)         message="wsmen:RenewResponseMessage"
(137)
wsa:Action="gost.e-ecolog.ru
(138)  wsam:Action="gost.e-ecolog.ru
(139)         />
(140)     </wsdl:operation>
(141)     <wsdl:operation name="GetStatusOp" >
(142)       <wsdl:input
(143)         message="wsmen:GetStatusMessage"
(144)
wsa:Action="gost.e-ecolog.ru
(145)
wsam:Action="gost.e-ecolog.ru
(146)         />
(147)       <wsdl:output
(148)         message="wsmen:GetStatusResponseMessage"
(149)
wsa:Action="gost.e-ecolog.ru
(150)  wsam:Action="gost.e-ecolog.ru
```

© ISO/IEC 2013 — All rights reserved

```
Version 1.1.1  DMTF Standard  195
(151)         />
(152)      </wsdl:operation>
(153)      <wsdl:operation name="ReleaseOp">
(154)        <wsdl:input
(155)          message="wsmen:ReleaseMessage"
(156)
wsa:Action="gost.e-ecolog.ru
(157)
wsam:Action="gost.e-ecolog.ru
(158)         />
(159)        <wsdl:output
(160)          message="wsmen:ReleaseResponseMessage"
(161)  wsa:Action="gost.e-ecolog.ru
(162)  wsam:Action="gost.e-ecolog.ru
(163)         />
(164)      </wsdl:operation>
(165)    </wsdl:portType>
(166)
(167)    <!-- The following portType shall be supported by the endpoint to which
(168)         The EnumerationEnd message is sent -->
(169)    <wsdl:portType name="EnumEndEndpoint">
(170)      <wsdl:operation name="EnumerationEndOp" >
(171)        <wsdl:input
(172)          message="wsmen:EnumerationEndMessage"
(173)
wsa:Action="gost.e-ecolog.ru
(174)  wsam:Action="gost.e-ecolog.ru
(175)         />
(176)      </wsdl:operation>
(177)    </wsdl:portType>
(178)  </wsdl:definitions>
```

Приложение I
(справочное)

XML-схема и WSDL операций уведомления

Нормативная копия XML-схем для операций уведомления может быть получена по следующему адресу: gost.e-ecolog.ru

Следующая ненормативная копия XML-схемы предоставлена для удобства:

```
(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!–
(3) DMTF — Distributed Management Task Force, Inc. — gost.e-ecolog.ru
(4)
(5) Document number: DSP8032
(6) Date: 2010-02-19
(7) Version: 1.0.0
(8) Document status: DMTF Standard
(9)
(10) Title: WS-Management Notification Operations XML Schema
(11)
(12) Document type: Specification (W3C XML Schema)
(13) Document language: E
(14)
(15) Abstract: XML Schema for WS-Management Notification Operations.
(16)
(17) Contact group: DMTF WS-Management Work Group, gost.e-ecolog.ru
(18)
(19) Copyright (C) 2008—2009 Distributed Management Task Force, Inc. (DMTF).
(20) All rights reserved.  DMTF is a not-for-profit association of industry
(21) members dedicated to promoting enterprise and systems management and
(22) interoperability.  Members and non-members may reproduce DMTF
(23) specifications and documents, provided that correct attribution is
(24) given.  As DMTF specifications may be revised from time to time,
(25) the particular version and release date should always be noted.
(26) Implementation of certain elements of this standard or proposed
(27) standard may be subject to third party patent rights, including
(28) provisional patent rights (herein "patent rights").  DMTF makes
(29) no representations to users of the standard as to the existence of
(30) such rights, and is not responsible to recognize, disclose,
(31) or identify any or all such third party patent right, owners or
(32) claimants, nor for any incomplete or inaccurate identification or
(33) disclosure of such rights, owners or claimants.  DMTF shall have no
(34) liability to any party, in any manner or circumstance, under any legal
(35) theory whatsoever, for failure to recognize, disclose, or identify any
(36) such third party patent rights, or for such party's reliance on the
(37) standard or incorporation thereof in its product, protocols or testing
(38) procedures.  DMTF shall have no liability to any party implementing
(39) such standard, whether such implementation is foreseeable or not, nor
(40) to any patent owner or claimant, and shall have no liability or
(41) responsibility for costs or losses incurred if a standard is withdrawn
```

```
(42) or modified after publication, and shall be indemnified and held
(43) harmless by any party implementing the standard from any and all claims
(44) of infringement by a patent owner for such implementations.  For
(45) information about patents held by third-parties which have notified the
(46) DMTF that, in their opinion, such patent may relate to or impact
(47) implementations of DMTF standards, visit
(48) gost.e-ecolog.ru
(49)
(50) Change log:
(51) 1.0.0 — 2009-11-01 — Work in Progress release
(52) 1.0.0 — 2010-02-19 — DMTF Standard release
(53)
(54)  -->
(55) <xs:schema
(56)  targetNamespace="gost.e-ecolog.ru
(57)  xmlns:tns="gost.e-ecolog.ru
(58)  xmlns:wsa="gost.e-ecolog.ru
(59)  xmlns:xs="gost.e-ecolog.ru
(60)  elementFormDefault="qualified"
(61)  blockDefault="#all">
(62)
(63)  <xs:import
(64)   namespace="gost.e-ecolog.ru
(65)   schemaLocation="gost.e-ecolog.ru                    />
(66)  <xs:import
(67)   namespace="gost.e-ecolog.ru
(68)   schemaLocation="gost.e-ecolog.ru                                    />
(69)  <xs:import
(70)   namespace="gost.e-ecolog.ru
(71)   schemaLocation="gost.e-ecolog.ru                                 />
(72)
(73)  <!-- Types and global elements -->
(74)  <xs:complexType name="DeliveryType" mixed="true">
(75)   <xs:sequence>
(76)    <xs:any namespace="##any" processContents="lax"
(77)          minOccurs="0" maxOccurs="unbounded" />
(78)   </xs:sequence>
(79)   <xs:attribute name="Mode" type="xs:anyURI" use="optional" />
(80)   <xs:anyAttribute namespace="##other" processContents="lax" />
(81)  </xs:complexType>
(82)
(83)  <xs:simpleType name="NonNegativeDurationType">
(84)   <xs:restriction base="xs:duration">
(85)    <xs:minInclusive value="P0Y0M0DT0H0M0S" />
(86)   </xs:restriction>
(87)  </xs:simpleType>
(88)
(89)  <xs:simpleType name="ExpirationType">
(90)    <xs:union memberTypes="xs:dateTime
```

```
(91)              tns:NonNegativeDurationType" />
(92)  </xs:simpleType>
(93)
(94)  <xs:complexType name="FilterType" mixed="true">
(95)    <xs:sequence>
(96)      <xs:any namespace="##other" processContents="lax"
(97)            minOccurs="0" maxOccurs="unbounded" />
(98)    </xs:sequence>
(99)    <xs:attribute name="Dialect" type="xs:anyURI" use="optional" />
(100)     <xs:anyAttribute namespace="##other" processContents="lax" />
(101)   </xs:complexType>
(102)
(103)   <xs:complexType name="LanguageSpecificStringType">
(104)     <xs:simpleContent>
(105)       <xs:extension base="xs:string">
(106)         <xs:attribute ref="xml:lang" />
(107)         <xs:anyAttribute namespace="##other" processContents="lax" />
(108)       </xs:extension>
(109)     </xs:simpleContent>
(110)   </xs:complexType>
(111)
(112)   <!--
(113)     The type of the AnyEPRType is effectively
(114)     the union of wsa04:EndpointReferenceType and
(115)     wsa10:EndpointReferenceType. Unfortunately, xs:union only
(116)     works for simple types. As a result, we have to define
(117)     the element in an unvalidated way to accommodate either
(118)     addressing type.
(119)     -->
(120)
(121)   <xs:complexType name="AnyEPRType">
(122)     <xs:sequence>
(123)       <xs:any minOccurs='1' maxOccurs='unbounded' processContents='skip'
(124)             namespace='##other' />
(125)     </xs:sequence>
(126)   </xs:complexType>
(127)
(128)   <xs:element name="NotifyTo" type="tns:AnyEPRType" />
(129)
(130)   <!-- Subscribe request -->
(131)   <xs:element name="Subscribe">
(132)     <xs:complexType>
(133)       <xs:sequence>
(134)         <xs:element name="EndTo" type="tns:AnyEPRType"
(135)                 minOccurs="0" />
(136)         <xs:element name="Delivery" type="tns:DeliveryType" />
(137)         <xs:element name="Expires" type="tns:ExpirationType"
(138)                 minOccurs="0" />
(139)         <xs:element name="Filter" type="tns:FilterType"
```

```
(140)                minOccurs="0" />
(141)       <xs:any namespace="##other" processContents="lax"
(142)             minOccurs="0" maxOccurs="unbounded" />
(143)     </xs:sequence>
(144)     <xs:anyAttribute namespace="##other" processContents="lax" />
(145)   </xs:complexType>
(146)  </xs:element>
(147)
(148)  <xs:element name="Identifier" type="xs:anyURI" />
(149)
(150)  <!-- Subscribe response -->
(151)  <xs:element name="SubscribeResponse">
(152)   <xs:complexType>
(153)     <xs:sequence>
(154)       <xs:element name="SubscriptionManager"
(155)             type="tns:AnyEPRType" />
(156)       <xs:element name="Expires" type="tns:ExpirationType" />
(157)       <xs:any namespace="##other" processContents="lax"
(158)             minOccurs="0" maxOccurs="unbounded" />
(159)     </xs:sequence>
(160)     <xs:anyAttribute namespace="##other" processContents="lax" />
(161)   </xs:complexType>
(162)  </xs:element>
(163)
(164)  <!-- Used in a fault if there's an unsupported dialect -->
(165)  <xs:element name="SupportedDialect" type="xs:anyURI" />
(166)
(167)  <!-- Used in a fault if there's an unsupported delivery mode -->
(168)  <xs:element name="SupportedDeliveryMode" type="xs:anyURI" />
(169)
(170)  <!-- Renew request -->
(171)  <xs:element name="Renew">
(172)   <xs:complexType>
(173)     <xs:sequence>
(174)       <xs:element name="Expires" type="tns:ExpirationType"
(175)             minOccurs="0" />
(176)       <xs:any namespace="##other" processContents="lax"
(177)             minOccurs="0" maxOccurs="unbounded" />
(178)     </xs:sequence>
(179)     <xs:anyAttribute namespace="##other" processContents="lax" />
(180)   </xs:complexType>
(181)  </xs:element>
(182)
(183)  <!-- Renew response -->
(184)  <xs:element name="RenewResponse">
(185)   <xs:complexType>
(186)     <xs:sequence>
(187)       <xs:element name="Expires" type="tns:ExpirationType"
(188)             minOccurs="0" />
```

```
(189)       <xs:any namespace="##other" processContents="lax"
(190)             minOccurs="0" maxOccurs="unbounded" />
(191)     </xs:sequence>
(192)     <xs:anyAttribute namespace="##other" processContents="lax" />
(193)   </xs:complexType>
(194)  </xs:element>
(195)
(196)  <!-- GetStatus request -->
(197)  <xs:element name="GetStatus">
(198)   <xs:complexType>
(199)     <xs:sequence>
(200)       <xs:any namespace="##other" processContents="lax"
(201)             minOccurs="0" maxOccurs="unbounded" />
(202)     </xs:sequence>
(203)     <xs:anyAttribute namespace="##other" processContents="lax" />
(204)   </xs:complexType>
(205)  </xs:element>
(206)
(207)  <!-- GetStatus response -->
(208)  <xs:element name="GetStatusResponse">
(209)   <xs:complexType>
(210)     <xs:sequence>
(211)       <xs:element name="Expires" type="tns:ExpirationType"
(212)             minOccurs="0" />
(213)       <xs:any namespace="##other" processContents="lax"
(214)             minOccurs="0" maxOccurs="unbounded" />
(215)     </xs:sequence>
(216)     <xs:anyAttribute namespace="##other" processContents="lax" />
(217)   </xs:complexType>
(218)  </xs:element>
(219)
(220)  <!-- Unsubscribe request -->
(221)  <xs:element name="Unsubscribe">
(222)   <xs:complexType>
(223)     <xs:sequence>
(224)       <xs:any namespace="##other" processContents="lax"
(225)             minOccurs="0" maxOccurs="unbounded" />
(226)     </xs:sequence>
(227)     <xs:anyAttribute namespace="##other" processContents="lax" />
(228)   </xs:complexType>
(229)  </xs:element>
(230)
(231)  <!-- SubscriptionEnd message -->
(232)  <xs:element name="SubscriptionEnd">
(233)   <xs:complexType>
(234)     <xs:sequence>
(235)       <xs:element name="SubscriptionManager"
(236)             type="tns:AnyEPRType" />
(237)       <xs:element name="Status"
```

```
(238)                type="tns:OpenSubscriptionEndCodeType" />
(239)       <xs:element name="Reason"
(240)                type="tns:LanguageSpecificStringType"
(241)                minOccurs="0" maxOccurs="unbounded" />
(242)       <xs:any namespace="##other" processContents="lax"
(243)            minOccurs="0" maxOccurs="unbounded" />
(244)     </xs:sequence>
(245)     <xs:anyAttribute namespace="##other" processContents="lax" />
(246)   </xs:complexType>
(247)  </xs:element>
(248)
(249)  <xs:simpleType name="SubscriptionEndCodeType">
(250)   <xs:restriction base="xs:anyURI">
(251)     <xs:enumeration
value="gost.e-ecolog.ru                                                    />
(252)     <xs:enumeration
value="gost.e-ecolog.ru                                                        />
(253)     <xs:enumeration
value="gost.e-ecolog.ru                                                    />
(254)   </xs:restriction>
(255)  </xs:simpleType>
(256)
(257)  <xs:simpleType name="OpenSubscriptionEndCodeType">
(258)   <xs:union memberTypes="tns:SubscriptionEndCodeType xs:anyURI" />
(259)  </xs:simpleType>
(260)
(261)  <xs:attribute name="EventSource" type="xs:boolean" />
(262) </xs:schema>
```

Нормативная копия описания WSDL может быть получена по адресу:

gost.e-ecolog.ru

Следующая ненормативная копия описания WSDL предоставлена для удобства:

```
(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!--
(3) DMTF — Distributed Management Task Force, Inc. — gost.e-ecolog.ru
(4)
(5) Document number: DSP8036
(6) Date: 2010-02-19
(7) Version: 1.0.0
(8) Document status: DMTF Standard
(9)
(10) Title: WS-Management Notification Operations WSDL
(11)
(12) Document type: Specification (W3C WSDL Document)
(13) Document language: E
(14)
(15) Abstract: WSDL for WS-Management Notification Operations.
(16)
(17) Contact group: DMTF WS-Management Work Group, gost.e-ecolog.ru
(18)
```

```
(19) Copyright (C) 2008–2010 Distributed Management Task Force, Inc. (DMTF).
(20) All rights reserved.  DMTF is a not-for-profit association of industry
(21) members dedicated to promoting enterprise and systems management and
(22) interoperability.  Members and non-members may reproduce DMTF
(23) specifications and documents, provided that correct attribution is
(24) given.  As DMTF specifications may be revised from time to time,
(25) the particular version and release date should always be noted.
(26) Implementation of certain elements of this standard or proposed
(27) standard may be subject to third party patent rights, including
(28) provisional patent rights (herein “patent rights”).  DMTF makes
(29) no representations to users of the standard as to the existence of
(30) such rights, and is not responsible to recognize, disclose,
(31) or identify any or all such third party patent right, owners or
(32) claimants, nor for any incomplete or inaccurate identification or
(33) disclosure of such rights, owners or claimants.  DMTF shall have no
(34) liability to any party, in any manner or circumstance, under any legal
(35) theory whatsoever, for failure to recognize, disclose, or identify any
(36) such third party patent rights, or for such party’s reliance on the
(37) standard or incorporation thereof in its product, protocols or testing
(38) procedures.  DMTF shall have no liability to any party implementing
(39) such standard, whether such implementation is foreseeable or not, nor
(40) to any patent owner or claimant, and shall have no liability or
(41) responsibility for costs or losses incurred if a standard is withdrawn
(42) or modified after publication, and shall be indemnified and held
(43) harmless by any party implementing the standard from any and all claims
(44) of infringement by a patent owner for such implementations.  For
(45) information about patents held by third-parties which have notified the
(46) DMTF that, in their opinion, such patent may relate to or impact
(47) implementations of DMTF standards, visit
(48) gost.e-ecolog.ru
(49)
(50) Change log:
(51) 1.0.0 — 2009-11-01 — Work in Progress release
(52) 1.0.0 — 2010-02-19 — DMTF Standard release
(53)
(54)  -->
(55) <wsdl:definitions
(56)  targetNamespace=“gost.e-ecolog.ru
(57)  xmlns:wsa=“gost.e-ecolog.ru
(58)  xmlns:wsam=“gost.e-ecolog.ru
(59)  xmlns:wsme=“gost.e-ecolog.ru
(60)  xmlns:wsdl=“gost.e-ecolog.ru
(61)  xmlns:xs=“gost.e-ecolog.ru                                    >
(62)
(63)  <wsdl:types>
(64)    <xs:schema>
(65)      <xs:import
(66)        namespace=“gost.e-ecolog.ru
(67)
```

```
schemaLocation="gost.e-ecolog.ru                                    />
(68)    </xs:schema>
(69)  </wsdl:types>
(70)
(71)  <wsdl:message name="SubscribeMsg" >
(72)    <wsdl:part name="body" element="wsme:Subscribe" />
(73)  </wsdl:message>
(74)  <wsdl:message name="SubscribeResponseMsg" >
(75)    <wsdl:part name="body" element="wsme:SubscribeResponse" />
(76)  </wsdl:message>
(77)
(78)  <wsdl:message name="RenewMsg" >
(79)    <wsdl:part name="body" element="wsme:Renew" />
(80)  </wsdl:message>
(81)  <wsdl:message name="RenewResponseMsg" >
(82)    <wsdl:part name="body" element="wsme:RenewResponse" />
(83)  </wsdl:message>
(84)
(85)  <wsdl:message name="GetStatusMsg" >
(86)    <wsdl:part name="body" element="wsme:GetStatus" />
(87)  </wsdl:message>
(88)  <wsdl:message name="GetStatusResponseMsg" >
(89)    <wsdl:part name="body" element="wsme:GetStatusResponse" />
(90)  </wsdl:message>
(91)
(92)  <wsdl:message name="UnsubscribeMsg" >
(93)    <wsdl:part name="body" element="wsme:Unsubscribe" />
(94)  </wsdl:message>
(95)  <wsdl:message name="UnsubscribeResponseMsg" />
(96)
(97)  <wsdl:message name="SubscriptionEnd" >
(98)    <wsdl:part name="body" element="wsme:SubscriptionEnd" />
(99)  </wsdl:message>
(100)
(101)   <wsdl:portType name="EventSource" >
(102)     <wsdl:operation name="SubscribeOp" >
(103)       <wsdl:input
(104)         message="wsme:SubscribeMsg"
(105)
wsa:Action="gost.e-ecolog.ru
(106)
wsam:Action="gost.e-ecolog.ru                                       >
(107)       <wsdl:output
(108)         message="wsme:SubscribeResponseMsg"
(109)
wsa:Action="gost.e-ecolog.ru
(110)
wsam:Action="gost.e-ecolog.ru                                                >
(111)     </wsdl:operation>
```

```
(112)   </wsdl:portType>
(113)
(114)   <!-- The following portType shall be supported by the endpoint to which
(115)        the SubscriptionEnd message is sent.  -->
(116)   <wsdl:portType name="EndToEndpoint">
(117)     <wsdl:operation name="SubscriptionEnd" >
(118)       <wsdl:input
(119)         message="wsme:SubscriptionEnd"
(120)
wsa:Action="gost.e-ecolog.ru
(121)
wsam:Action="gost.e-ecolog.ru                                                    >
(122)     </wsdl:operation>
(123)   </wsdl:portType>
(124)
(125)   <!-- The following portType shall be supported by the endpoint to which
(126)        Notifications are sent.  This portType also serves as a
(127)        mechanism by which Subscribers can know the Notifications that
(128)        will sent by an Event Source. -->
(129)   <wsdl:portType name="EventSink">
(130)     <!-- place the Notification messages (operations) here. For example:
(131)     <wsdl:operation name="WeatherReport">
(132)     <wsdl:input message="wr:ThunderStormMessage"
(133)       wsa:Action="urn:weatherReport:ThunderStorm"
(134)       wsam:Action="urn:weatherReport:ThunderStorm" />
(135)   </wsdl:operation>
(136)   -->
(137)   </wsdl:portType>
(138)
(139)   <wsdl:portType name="SubscriptionManager" >
(140)     <wsdl:operation name="RenewOp" >
(141)       <wsdl:input
(142)         message="wsme:RenewMsg"
(143)         wsa:Action="gost.e-ecolog.ru
(144)
wsam:Action="gost.e-ecolog.ru                                                >
(145)       <wsdl:output
(146)         message="wsme:RenewResponseMsg"
(147)
wsa:Action="gost.e-ecolog.ru
(148)
wsam:Action="gost.e-ecolog.ru                                                    >
(149)     </wsdl:operation>
(150)     <wsdl:operation name="GetStatusOp" >
(151)       <wsdl:input
(152)         message="wsme:GetStatusMsg"
(153)
wsa:Action="gost.e-ecolog.ru
(154)
```

```
wsam:Action="gost.e-ecolog.ru                                                    >
(155)       <wsdl:output
(156)         message="wsme:GetStatusResponseMsg"
(157)
wsa:Action="gost.e-ecolog.ru
(158)
wsam:Action="gost.e-ecolog.ru                                                          >
(159)     </wsdl:operation>
(160)     <wsdl:operation name="UnsubscribeOp" >
(161)       <wsdl:input
(162)         message="wsme:UnsubscribeMsg"
(163)
wsa:Action="gost.e-ecolog.ru
(164)
wsam:Action="gost.e-ecolog.ru                                                     >
(165)       <wsdl:output
(166)         message="wsme:UnsubscribeResponseMsg"
(167)
wsa:Action="gost.e-ecolog.ru
(168)
wsam:Action="gost.e-ecolog.ru                                                           >
(169)     </wsdl:operation>
(170)   </wsdl:portType>
(171) </wsdl:definitions>
```

Приложение J
(справочное)

XML-схема адресации

Нормативная копия XML-схем для функции адресации может быть получена по следующему адресу: gost.e-ecolog.ru

Следующая ненормативная копия XML-схемы предоставлена для удобства:

```
(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!--
(3) DMTF — Distributed Management Task Force, Inc. — gost.e-ecolog.ru
(4)
(5) Document number: DSP8034
(6) Date: 2010-02-19
(7) Version: 1.0.0
(8) Document status: DMTF Standard
(9)
(10) Title: WS-Management Addressing XML Schema
(11)
(12) Document type: Specification (W3C XML Schema)
(13) Document language: E
(14)
(15) Abstract: XML Schema for WS-Management Addressing.
(16)
(17) Contact group: DMTF WS-Management Work Group, gost.e-ecolog.ru
(18)
(19) Copyright (C) 2008—2010 Distributed Management Task Force, Inc. (DMTF).
(20) All rights reserved. DMTF is a not-for-profit association of industry
(21) members dedicated to promoting enterprise and systems management and
(22) interoperability. Members and non-members may reproduce DMTF
(23) specifications and documents, provided that correct attribution is
(24) given. As DMTF specifications may be revised from time to time,
(25) the particular version and release date should always be noted.
(26) Implementation of certain elements of this standard or proposed
(27) standard may be subject to third party patent rights, including
(28) provisional patent rights (herein "patent rights"). DMTF makes
(29) no representations to users of the standard as to the existence of
(30) such rights, and is not responsible to recognize, disclose,
(31) or identify any or all such third party patent right, owners or
(32) claimants, nor for any incomplete or inaccurate identification or
(33) disclosure of such rights, owners or claimants. DMTF shall have no
(34) liability to any party, in any manner or circumstance, under any legal
(35) theory whatsoever, for failure to recognize, disclose, or identify any
(36) such third party patent rights, or for such party's reliance on the
(37) standard or incorporation thereof in its product, protocols or testing
(38) procedures. DMTF shall have no liability to any party implementing
(39) such standard, whether such implementation is foreseeable or not, nor
(40) to any patent owner or claimant, and shall have no liability or
(41) responsibility for costs or losses incurred if a standard is withdrawn
```

```
(42) or modified after publication, and shall be indemnified and held
(43) harmless by any party implementing the standard from any and all claims
(44) of infringement by a patent owner for such implementations.  For
(45) information about patents held by third-parties which have notified the
(46) DMTF that, in their opinion, such patent may relate to or impact
(47) implementations of DMTF standards, visit
(48) gost.e-ecolog.ru
(49)
(50) Change log:
(51) 1.0.0 — 2009-11-01 — Work in Progress release
(52) 1.0.0 — 2010-02-19 — DMTF Standard release
(53)  -->
(54) <xs:schema
(55)  targetNamespace="gost.e-ecolog.ru
(56)  xmlns:xs="gost.e-ecolog.ru
(57)  xmlns:wsa="gost.e-ecolog.ru
(58)  elementFormDefault="qualified" blockDefault="#all">
(59)
(60)  <!-- ////////////////////// Addressing ////////////////////// -->
(61)  <!-- Endpoint reference -->
(62)  <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
(63)  <xs:complexType name="EndpointReferenceType">
(64)   <xs:sequence>
(65)    <xs:element name="Address" type="wsa:AttributedURI"/>
(66)    <xs:element name="ReferenceProperties"
(67)     type="wsa:ReferencePropertiesType" minOccurs="0"/>
(68)    <xs:element name="ReferenceParameters"
(69)     type="wsa:ReferenceParametersType" minOccurs="0"/>
(70)    <xs:element name="PortType" type="wsa:AttributedQName"  minOccurs="0"/>
(71)    <xs:element name="ServiceName" type="wsa:ServiceNameType"  minOccurs="0"/>
(72)    <xs:any namespace="##other" processContents="lax" minOccurs="0"
(73)     maxOccurs="unbounded">
(74)     <xs:annotation>
(75)      <xs:documentation>
(76)       If "Policy" elements from namespace
(77)       "gost.e-ecolog.ru                                    are used,
(78)       they must appear first (before any extensibility elements).
(79)      </xs:documentation>
(80)     </xs:annotation>
(81)    </xs:any>
(82)   </xs:sequence>
(83)   <xs:anyAttribute namespace="##other" processContents="lax"/>
(84)  </xs:complexType>
(85)  <xs:complexType name="ReferencePropertiesType">
(86)   <xs:sequence>
(87)    <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
(88)   </xs:sequence>
(89)  </xs:complexType>
(90)  <xs:complexType name="ReferenceParametersType">
```

```
(91)    <xs:sequence>
(92)      <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
(93)    </xs:sequence>
(94)  </xs:complexType>
(95)  <xs:complexType name="ServiceNameType">
(96)    <xs:simpleContent>
(97)      <xs:extension base="xs:QName">
(98)        <xs:attribute name="PortName" type="xs:NCName"/>
(99)        <xs:anyAttribute namespace="##other" processContents="lax"/>
(100)     </xs:extension>
(101)   </xs:simpleContent>
(102) </xs:complexType>
(103) <!-- Message information header blocks -->
(104) <xs:element name="MessageID" type="wsa:AttributedURI"/>
(105) <xs:element name="RelatesTo" type="wsa:Relationship"/>
(106) <xs:element name="To" type="wsa:AttributedURI"/>
(107) <xs:element name="Action" type="wsa:AttributedURI"/>
(108) <xs:element name="From" type="wsa:EndpointReferenceType"/>
(109) <xs:element name="ReplyTo" type="wsa:EndpointReferenceType"/>
(110) <xs:element name="FaultTo" type="wsa:EndpointReferenceType"/>
(111) <xs:complexType name="Relationship">
(112)   <xs:simpleContent>
(113)     <xs:extension base="xs:anyURI">
(114)       <xs:attribute name="RelationshipType" type="xs:QName" use="optional"/>
(115)       <xs:anyAttribute namespace="##other" processContents="lax"/>
(116)     </xs:extension>
(117)   </xs:simpleContent>
(118) </xs:complexType>
(119) <xs:simpleType name="RelationshipTypeValues">
(120)   <xs:restriction base="xs:QName">
(121)     <xs:enumeration value="wsa:Reply"/>
(122)   </xs:restriction>
(123) </xs:simpleType>
(124) <xs:element name="ReplyAfter" type="wsa:ReplyAfterType"/>
(125) <xs:complexType name="ReplyAfterType">
(126)   <xs:simpleContent>
(127)     <xs:extension base="xs:nonNegativeInteger">
(128)       <xs:anyAttribute namespace="##other"/>
(129)     </xs:extension>
(130)   </xs:simpleContent>
(131) </xs:complexType>
(132) <xs:element name="RetryAfter" type="wsa:RetryAfterType"/>
(133) <xs:complexType name="RetryAfterType">
(134)   <xs:simpleContent>
(135)     <xs:extension base="xs:nonNegativeInteger">
(136)       <xs:anyAttribute namespace="##other"/>
(137)     </xs:extension>
(138)   </xs:simpleContent>
(139) </xs:complexType>
```

```
(140)   <xs:simpleType name="FaultSubcodeValues">
(141)     <xs:restriction base="xs:QName">
(142)       <xs:enumeration value="wsa:InvalidMessageInformationHeader"/>
(143)       <xs:enumeration value="wsa:MessageInformationHeaderRequired"/>
(144)       <xs:enumeration value="wsa:DestinationUnreachable"/>
(145)       <xs:enumeration value="wsa:ActionNotSupported"/>
(146)       <xs:enumeration value="wsa:EndpointUnavailable"/>
(147)     </xs:restriction>
(148)   </xs:simpleType>
(149)   <xs:attribute name="Action" type="xs:anyURI"/>
(150)   <!-- Common declarations and definitions -->
(151)   <xs:complexType name="AttributedQName">
(152)     <xs:simpleContent>
(153)       <xs:extension base="xs:QName">
(154)         <xs:anyAttribute namespace="##other" processContents="lax"/>
(155)       </xs:extension>
(156)     </xs:simpleContent>
(157)   </xs:complexType>
(158)   <xs:complexType name="AttributedURI">
(159)     <xs:simpleContent>
(160)       <xs:extension base="xs:anyURI">
(161)         <xs:anyAttribute namespace="##other" processContents="lax"/>
(162)       </xs:extension>
(163)     </xs:simpleContent>
(164)   </xs:complexType>
(165) </xs:schema>
```

Приложение К
(справочное)

XML-схема WS-Management

Нормативная копия XML-схем для WS-Management может быть получена по следующему адресу:
gost.e-ecolog.ru
Следующая ненормативная копия XML-схемы предоставлена для удобства:

(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <!--
(3) Notice
(4) DSP8015
(5) Document: WS-Management protocol XML Schema
(6) Version: 1.0.0
(7) Status: Final
(8) Date: 01/20/2008
(9) Author: Bryan Murray, et al.
(10) Description: XML Schema for WS-Management protocol
(11)
(12) Copyright © 2008 Distributed Management Task Force, Inc. (DMTF). All rights reserved. DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted. Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations. For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit gost.e-ecolog.ru
(13)
(14) Change Requests:
(15) None
(16) -->

```
(17) <xs:schema targetNamespace="gost.e-ecolog.ru
(18)   xmlns:wsman="gost.e-ecolog.ru
(19)   xmlns:wsa="gost.e-ecolog.ru
(20)   xmlns:xs="gost.e-ecolog.ru
(21)   elementFormDefault="qualified" version="1.0.0e">
(22)
(23)  <xs:import namespace="gost.e-ecolog.ru
(24)
schemaLocation="gost.e-ecolog.ru                                    >
(25)  <xs:import namespace="gost.e-ecolog.ru
(26)          schemaLocation="gost.e-ecolog.ru                      >
(27)
(28)  <xs:complexType name="attributableURI">
(29)   <xs:simpleContent>
(30)    <xs:extension base="xs:anyURI">
(31)     <xs:anyAttribute namespace="##other" processContents="lax"/>
(32)    </xs:extension>
(33)   </xs:simpleContent>
(34)  </xs:complexType>
(35)
(36)  <xs:element name="ResourceURI" type="wsman:attributableURI"/>
(37)
(38)  <xs:complexType name="SelectorType">
(39)   <xs:annotation>
(40)    <xs:documentation>
(41)     Instances of this type can be only simple types or EPRs, not arbitrary mixed data.
(42)    </xs:documentation>
(43)   </xs:annotation>
(44)   <xs:complexContent mixed="true">
(45)    <xs:restriction base="xs:anyType">
(46)     <xs:sequence>
(47)      <xs:element ref="wsa:EndpointReference" minOccurs="0"/>
(48)     </xs:sequence>
(49)     <xs:attribute name="Name" type="xs:NCName" use="required"/>
(50)     <xs:anyAttribute namespace="##other" processContents="lax"/>
(51)    </xs:restriction>
(52)   </xs:complexContent>
(53)  </xs:complexType>
(54)  <xs:element name="Selector" type="wsman:SelectorType"/>
(55)
(56)  <xs:complexType name="SelectorSetType">
(57)   <xs:sequence>
(58)    <xs:element ref="wsman:Selector" minOccurs="1" maxOccurs="unbounded"/>
(59)   </xs:sequence>
(60)   <xs:anyAttribute namespace="##other" processContents="lax"/>
(61)  </xs:complexType>
(62)
(63)  <xs:element name="SelectorSet" type="wsman:SelectorSetType">
(64)   <xs:unique name="oneSelectorPerName">
```

```
(65)      <xs:selector xpath=“./Selector”/>
(66)      <xs:field xpath=“@Name”/>
(67)    </xs:unique>
(68)  </xs:element>
(69)
(70)  <xs:complexType name=“attributableDuration”>
(71)    <xs:simpleContent>
(72)      <xs:extension base=“xs:duration”>
(73)        <xs:anyAttribute namespace=“##other” processContents=“lax”/>
(74)      </xs:extension>
(75)    </xs:simpleContent>
(76)  </xs:complexType>
(77)
(78)  <xs:element name=“OperationTimeout” type=“wsman:attributableDuration”/>
(79)
(80)  <xs:complexType name=“attributablePositiveInteger”>
(81)    <xs:simpleContent>
(82)      <xs:extension base=“xs:positiveInteger”>
(83)        <xs:anyAttribute namespace=“##other” processContents=“lax”/>
(84)      </xs:extension>
(85)    </xs:simpleContent>
(86)  </xs:complexType>
(87)
(88)  <xs:simpleType name=“PolicyType”>
(89)    <xs:restriction base=“xs:token”>
(90)      <xs:enumeration value=“CancelSubscription”/>
(91)      <xs:enumeration value=“Skip”/>
(92)      <xs:enumeration value=“Notify”/>
(93)    </xs:restriction>
(94)  </xs:simpleType>
(95)
(96)  <xs:complexType name=“MaxEnvelopeSizeType”>
(97)    <xs:simpleContent>
(98)      <xs:extension base=“wsman:attributablePositiveInteger”>
(99)        <xs:attribute name=“Policy” type=“wsman:PolicyType” default=“Notify”/>
(100)     </xs:extension>
(101)   </xs:simpleContent>
(102) </xs:complexType>
(103) <xs:element name=“MaxEnvelopeSize” type=“wsman:MaxEnvelopeSizeType”/>
(104)
(105) <xs:element name=“Locale”>
(106)   <xs:complexType>
(107)     <xs:attribute ref=“xml:lang” use=“required”/>
(108)     <xs:anyAttribute namespace=“##other” processContents=“lax”/>
(109)   </xs:complexType>
(110) </xs:element>
(111)
(112) <xs:complexType name=“OptionType”>
(113)   <xs:simpleContent>
```

```
(114)     <xs:extension base="xs:string">
(115)       <xs:attribute name="Name" type="xs:NCName" use="required"/>
(116)       <xs:attribute name="MustComply" type="xs:boolean" default="false"/>
(117)       <xs:attribute name="Type" type="xs:QName"/>
(118)       <xs:anyAttribute namespace="##other" processContents="lax"/>
(119)     </xs:extension>
(120)   </xs:simpleContent>
(121) </xs:complexType>
(122) <xs:element name="Option" type="wsman:OptionType"/>
(123)
(124) <xs:element name="OptionSet">
(125)   <xs:complexType>
(126)     <xs:sequence>
(127)       <xs:element ref="wsman:Option" minOccurs="0" maxOccurs="unbounded"/>
(128)     </xs:sequence>
(129)     <xs:anyAttribute namespace="##other" processContents="lax"/>
(130)   </xs:complexType>
(131) </xs:element>
(132)
(133) <xs:complexType name="attributableEmpty">
(134)   <xs:anyAttribute namespace="##other" processContents="lax"/>
(135) </xs:complexType>
(136)
(137) <xs:element name="RequestEPR" type="wsman:attributableEmpty"/>
(138) <xs:element name="EPRInvalid" type="wsman:attributableEmpty"/>
(139) <xs:element name="EPRUnknown" type="wsman:attributableEmpty"/>
(140)
(141) <xs:complexType name="RequestedEPRType">
(142)   <xs:choice>
(143)     <xs:element ref="wsa:EndpointReference"/>
(144)     <xs:element ref="wsman:EPRInvalid"/>
(145)     <xs:element ref="wsman:EPRUnknown"/>
(146)   </xs:choice>
(147)   <xs:anyAttribute namespace="##other" processContents="lax"/>
(148) </xs:complexType>
(149) <xs:element name="RequestedEPR" type="wsman:RequestedEPRType"/>
(150)
(151) <xs:complexType name="mixedDataType">
(152)   <xs:complexContent mixed="true">
(153)     <xs:restriction base="xs:anyType">
(154)       <xs:sequence>
(155)         <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="skip"/>
(156)       </xs:sequence>
(157)       <xs:anyAttribute namespace="##other" processContents="lax"/>
(158)     </xs:restriction>
(159)   </xs:complexContent>
(160) </xs:complexType>
(161)
(162) <xs:complexType name="fragmentMixedDataType">
```

```
(163)    <xs:complexContent mixed="true">
(164)      <xs:extension base="wsman:mixedDataType">
(165)  <xs:attribute name="Dialect" type="xs:anyURI" default="
(166)        <xs:anyAttribute namespace="##other" processContents="lax"/>
(167)      </xs:extension>
(168)    </xs:complexContent>
(169)  </xs:complexType>
(170)
(171)  <xs:element name="FragmentTransfer" type="wsman:fragmentMixedDataType"/>
(172)  <xs:element name="XmlFragment" type="wsman:mixedDataType"/>
(173)
(174)  <xs:complexType name="attributableNonNegativeInteger">
(175)    <xs:simpleContent>
(176)      <xs:extension base="xs:nonNegativeInteger">
(177)        <xs:anyAttribute namespace="##other" processContents="lax"/>
(178)      </xs:extension>
(179)    </xs:simpleContent>
(180)  </xs:complexType>
(181)
(182)  <xs:element name="TotalItemsCountEstimate"
type="wsman:attributableNonNegativeInteger" nillable="true"/>
(183)  <xs:element name="RequestTotalItemsCountEstimate"  type="wsman:attributableEmpty"/>
(184)
(185)  <xs:element name="OptimizeEnumeration" type="wsman:attributableEmpty"/>
(186)  <xs:element name="MaxElements" type="wsman:attributablePositiveInteger"/>
(187)
(188)  <xs:simpleType name="EnumerationModeType">
(189)    <xs:restriction base="xs:token">
(190)      <xs:enumeration value="EnumerateEPR"/>
(191)      <xs:enumeration value="EnumerateObjectAndEPR"/>
(192)    </xs:restriction>
(193)  </xs:simpleType>
(194)  <xs:element name="EnumerationMode" type="wsman:EnumerationModeType"/>
```

© ISO/IEC 2013 — All rights reserved ISO/IEC 17963:2013(E)


```
(195)
(196)  <xs:complexType name="mixedDataFilterType" mixed="true">
(197)    <xs:complexContent mixed="true">
(198)      <xs:restriction base="xs:anyType">
(199)        <xs:sequence>
(200)          <xs:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
(201)        </xs:sequence>
(202)        <xs:anyAttribute namespace="##any" processContents="lax"/>
(203)      </xs:restriction>
(204)    </xs:complexContent>
(205)  </xs:complexType>
(206)
(207)  <xs:complexType name="filterMixedDataType" mixed="true">
```

```
(208)   <xs:complexContent mixed="true">
(209)     <xs:extension base="wsman:mixedDataFilterType">
(210)  <xs:attribute name="Dialect" type="xs:anyURI" default="gost.e-ecolog.ru                    /1999/REC-xpath-19991116"/>
(211)       <xs:anyAttribute namespace="##any" processContents="lax"/>
(212)     </xs:extension>
(213)   </xs:complexContent>
(214)  </xs:complexType>
(215)
(216)  <xs:element name="Filter" type="wsman:filterMixedDataType"/>
(217)
(218)  <xs:complexType name="ObjectAndEPRType">
(219)   <xs:sequence>
(220)     <xs:any namespace="##any" processContents="lax"/>
(221)     <xs:element ref="wsa:EndpointReference"/>
(222)   </xs:sequence>
(223)  </xs:complexType>
(224)  <xs:element name="Item" type="wsman:ObjectAndEPRType"/>
(225)
(226)  <xs:complexType name="anyListType">
(227)   <xs:sequence>
(228)     <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
(229)   </xs:sequence>
(230)   <xs:anyAttribute namespace="##other" processContents="lax"/>
(231)  </xs:complexType>
(232)
(233)  <xs:element name="Items" type="wsman:anyListType"/>
(234)  <xs:element name="EndOfSequence" type="wsman:attributableEmpty"/>
(235)
(236)  <xs:complexType name="attributableLanguage">
(237)   <xs:simpleContent>
(238)     <xs:extension base="xs:language">
(239)       <xs:anyAttribute namespace="##other" processContents="lax"/>
(240)     </xs:extension>
(241)   </xs:simpleContent>
(242)  </xs:complexType>
(243)
(244)  <xs:element name="ContentEncoding" type="wsman:attributableLanguage"/>
(245)
(246)  <xs:complexType name="ConnectionRetryType">
(247)   <xs:simpleContent>
(248)     <xs:extension base="wsman:attributableDuration">
(249)       <xs:attribute name="Total" type="xs:unsignedLong"/>
(250)     </xs:extension>
(251)   </xs:simpleContent>
(252)  </xs:complexType>
(253)  <xs:element name="ConnectionRetry" type="wsman:ConnectionRetryType"/>
(254)
(255)  <xs:element name="Heartbeats" type="wsman:attributableDuration"/>
(256)  <xs:element name="SendBookmarks" type="wsman:attributableEmpty"/>
```

```
(257)
(258)  <xs:complexType name="attributableAny">
(259)   <xs:sequence>
(260)    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
(261)   </xs:sequence>
(262)   <xs:anyAttribute namespace="##other" processContents="lax"/>
(263)  </xs:complexType>
(264)
(265)  <xs:element name="Bookmark" type="wsman:mixedDataType"/>
(266)  <xs:element name="MaxTime" type="wsman:attributableDuration"/>
(267)
(268)  <xs:complexType name="EventType">
(269)   <xs:complexContent>
(270)    <xs:extension base="wsman:attributableAny">
(271)     <xs:attribute name="Action" type="xs:anyURI" use="required"/>
(272)    </xs:extension>
(273)   </xs:complexContent>
(274)  </xs:complexType>
(275)  <xs:element name="Event" type="wsman:EventType"/>
(276)
(277)  <xs:complexType name="EventsType">
(278)   <xs:sequence>
(279)    <xs:element ref="wsman:Event" minOccurs="1" maxOccurs="unbounded"/>
(280)   </xs:sequence>
(281)   <xs:anyAttribute namespace="##other" processContents="lax"/>
(282)  </xs:complexType>
(283)  <xs:element name="Events" type="wsman:EventsType"/>
(284)
(285)  <xs:element name="AckRequested" type="wsman:attributableEmpty"/>
(286)
(287)  <xs:complexType name="attributableInt">
(288)   <xs:simpleContent>
(289)    <xs:extension base="xs:int">
(290)     <xs:anyAttribute namespace="##other" processContents="lax"/>
(291)    </xs:extension>
(292)   </xs:simpleContent>
(293)  </xs:complexType>
(294)
(295)  <xs:complexType name="DroppedEventsType">
(296)   <xs:simpleContent>
(297)    <xs:extension base="wsman:attributableInt">
(298)     <xs:attribute name="Action" type="xs:anyURI" use="required"/>
(299)    </xs:extension>
(300)   </xs:simpleContent>
(301)  </xs:complexType>
(302)  <xs:element name="DroppedEvents" type="wsman:DroppedEventsType"/>
(303)
(304)  <xs:simpleType name="restrictedProfileType">
(305)   <xs:restriction base="xs:anyURI">
```

```
(306)     <xs:enumeration
value="gost.e-ecolog.ru                                          >
(307)     <xs:enumeration
value="gost.e-ecolog.ru                                           >
(308)     <xs:enumeration
value="gost.e-ecolog.ru                                           >
(309)     <xs:enumeration
value="gost.e-ecolog.ru                                            >
(310)     <xs:enumeration
value="gost.e-ecolog.ru                                             >
(311)     <xs:enumeration
value="gost.e-ecolog.ru                                                  "/>
(312)     <xs:enumeration
value="gost.e-ecolog.ru                                                  t"/>
(313)     <xs:enumeration
value="gost.e-ecolog.ru                                                     >
(314)     <xs:enumeration
value="gost.e-ecolog.ru                                                        >
(315)     <xs:enumeration
value="gost.e-ecolog.ru                                                     >
(316)    </xs:restriction>
(317)   </xs:simpleType>
(318)
(319)   <xs:simpleType name="ProfileType">
(320)    <xs:union memberTypes="wsman:restrictedProfileType xs:anyURI"/>
(321)   </xs:simpleType>
(322)
(323)   <xs:complexType name="AuthType">
(324)    <xs:complexContent>
(325)     <xs:extension base="wsman:attributableEmpty">
(326)      <xs:attribute name="Profile" type="wsman:ProfileType" use="required"/>
(327)     </xs:extension>
(328)    </xs:complexContent>
(329)   </xs:complexType>
(330)   <xs:element name="Auth" type="wsman:AuthType"/>
(331)
(332)   <xs:simpleType name="ThumbprintType">
(333)     <xs:restriction base="xs:string">
(334)       <xs:pattern value="[0-9a-fA-F]{40}"/>
(335)     </xs:restriction>
(336)   </xs:simpleType>
(337)   <xs:element name="CertificateThumbprint" type="wsman:ThumbprintType"/>
(338)
(339)
(340)   <xs:simpleType name="restrictedFaultDetailType">
(341)    <xs:restriction base="xs:anyURI">
(342)     <xs:enumeration
value="gost.e-ecolog.ru                                                  >
(343)     <xs:enumeration
```

value=“ gost.e-ecolog.ru >
(344) <xs:enumeration
value=“ gost.e-ecolog.ru >
(345) <xs:enumeration
value=“ gost.e-ecolog.ru >
(346) <xs:enumeration
value=“ gost.e-ecolog.ru >
(347) <xs:enumeration
value=“ gost.e-ecolog.ru >
(348) <xs:enumeration
value=“ gost.e-ecolog.ru >
(349) <xs:enumeration
value=“ gost.e-ecolog.ru >
(350) <xs:enumeration
value=“ gost.e-ecolog.ru >
(351) <xs:enumeration
value=“ gost.e-ecolog.ru >
(352) <xs:enumeration
value=“ gost.e-ecolog.ru >
(353) <xs:enumeration
value=“ gost.e-ecolog.ru >
(354) <xs:enumeration
value=“ gost.e-ecolog.ru >
(355) <xs:enumeration
value=“ gost.e-ecolog.ru >
(356) <xs:enumeration
value=“ gost.e-ecolog.ru >
(357) <xs:enumeration
value=“ gost.e-ecolog.ru >
(358) <xs:enumeration
value=“ gost.e-ecolog.ru >
(359) <xs:enumeration
value=“ gost.e-ecolog.ru >
(360) <xs:enumeration
value=“ gost.e-ecolog.ru >
(361) <xs:enumeration
value=“ gost.e-ecolog.ru >
(362) <xs:enumeration
value=“ gost.e-ecolog.ru >
(363) <xs:enumeration
value=“ gost.e-ecolog.ru >
(364) <xs:enumeration
value=“ gost.e-ecolog.ru >
(365) <xs:enumeration
value=“ gost.e-ecolog.ru >
(366) <xs:enumeration
value=“ gost.e-ecolog.ru >
(367) <xs:enumeration
value=“ gost.e-ecolog.ru >

```
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
<xs:enumeration
value="gost.e-ecolog.ru >
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="FaultDetailType">
<xs:union memberTypes="wsman:restrictedFaultDetailType xs:anyURI"/>
</xs:simpleType>

<xs:element name="FaultDetail" type="wsman:FaultDetailType"/>
<xs:element name="FragmentDialect" type="wsman:attributableURI"/>
<xs:element name="SupportedSelectorName" type="xs:NCName"/>
```

```
(398)  <!-- Master Fault Table subcode QNames -->
(399)  <xs:element name="AccessDenied"><xs:complexType/></xs:element>
(400)  <xs:element name="AlreadyExists"><xs:complexType/></xs:element>
(401)  <xs:element name="CannotProcessFilter"><xs:complexType/></xs:element>
(402)  <xs:element name="Concurrency"><xs:complexType/></xs:element>
(403)  <xs:element name="DeliveryRefused"><xs:complexType/></xs:element>
(404)  <xs:element name="EncodingLimit"><xs:complexType/></xs:element>
(405)  <xs:element name="EventDeliverToUnusable"><xs:complexType/></xs:element>
(406)  <xs:element name="FragmentDialectNotSupported"><xs:complexType/></xs:element>
(407)  <xs:element name="InternalError"><xs:complexType/></xs:element>
(408)  <xs:element name="InvalidBookmark"><xs:complexType/></xs:element>
(409)  <xs:element name="InvalidOptions"><xs:complexType/></xs:element>
(410)  <xs:element name="InvalidParameter"><xs:complexType/></xs:element>
(411)  <xs:element name="InvalidSelectors"><xs:complexType/></xs:element>
(412)  <xs:element name="NoAck"><xs:complexType/></xs:element>
(413)  <xs:element name="QuotaLimit"><xs:complexType/></xs:element>
(414)  <xs:element name="SchemaValidationError"><xs:complexType/></xs:element>
(415)  <xs:element name="TimedOut"><xs:complexType/></xs:element>
(416)  <xs:element name="UnsupportedFeature"><xs:complexType/></xs:element>
(417)
(418) </xs:schema>
```

Приложение L
(справочное)

Список изменений

Версия	Дата	Описание
1.0.0	2008-02-12	Опубликован как заключительный стандарт
1.1.0	2010-03-03	Опубликован как стандарт DMTF, со следующими изменениями: - в текст включены спецификации TEEN - решены противоречия с DSP0227 для Put и FragmentPut
1.1.1	2012-07-30	Добавлен необязательный поясняющий текст в предисловие для публикации ISO/IEC как Общедоступная спецификация (PAS)
1.1.1	2012-08-28	Стандарт DMTF

Приложение ДА
(справочное)

Сведение о соответствии ссылочного международного стандарта национальному стандарту

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
The Unicode Consortium, The Unicode Standard Version 3.0	—	* Стандарт Юникод версия 3.0
* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта.		

УДК 004.057.9:006.354 ОКС 35.020

Ключевые слова: спецификация веб-служб, управляемый ресурс, уведомление

Редактор *М.Ю. Сухинина*
Технический редактор *В.Н. Прусакова*
Корректор *М.В. Бучная*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 25.05.2016. Подписано в печать 20.06.2016. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 23,72. Уч.-изд. л. 20,00.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Издано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru