ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 17203— 2013

Информационные технологии
СПЕЦИФИКАЦИЯ ОТКРЫТОГО ФОРМАТА ВИЗУАЛИЗАЦИИ (OVF)

ISO/IEC 17203:2011
Information technology – Open Virtualization Format (OVF) specification
(IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Информационный аналитический вычислительный центр» (ООО «ИАВЦ») на основе собственного аутентичного перевода на русский язык 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 «08» ноября 2013 № 1542-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 17203:2011 «Информационная технология. Спецификация открытого формата визуализации (OVF)» (ISO/IEC 17203:2011 «Information technology – Open Virtualization Format (OVF) specification»).

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5 (пункт 3.5).

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 (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок – в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске информационного указателя «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.ru)

© Стандартинформ, 2014

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 ... 1
3 Термины и определения ... 2
4 Условные обозначения и сокращения ... 3
5 Пакеты OVF ... 3
5.1 Структура пакета OVF ... 3
5.2 Виртуальные дисковые форматы ... 5
5.3 Распространение в виде одного файла ... 5
5.4 Распространение в виде несольких файлов ... 5
6 Дескриптор OVF ... 6
7 Элемент конверт (Envelope) ... 6
7.1 Файловые ссылки (References) ... 7
7.2 Элемент контента (Content) ... 8
7.3 Возможности расширения ... 9
7.4 Соответствие ... 10
8 Описание виртуального аппаратного обеспечения ... 11
8.1 Раздел виртуальных аппаратных средств VirtualHardwareSection ... 11
8.2 Возможности расширения ... 12
8.3 Элементы виртуального аппаратног обеспечения ... 12
8.4 Диапазоны элементов ... 15
9 Основные разделы метаданных ... 16
9.1 Раздел дисков DiskSection ... 16
9.2 Раздел сетей NetworkSection ... 18
9.3 Раздел выделения ресурсов ResourceAllocationSection ... 18
9.4 Раздел аннотации AnnotationSection ... 19
9.5 Раздел продукта ProductSection ... 19
9.6 Раздел лицензий EulaSection ... 22
9.7 Раздел запуска StartupSection ... 22
9.8 Раздел опций развертывания DeploymentOptionSection ... 23
9.9 Раздел операционной системы OperatingSystemSection ... 25
9.10 Раздел установки InstallSection ... 25
10 Интернационализация ... 25
11 Среда OVF (OVF Environment) ... 27
11.1 Документ среды Environment Document ... 27
11.2 Раздел транспорта (Transport) ... 28
Приложение А (информативное) ... 30
Приложение В (информативное) ... 31
Приложение С (нормативное) ... 32
Библиография ... 33
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации ... 34

Введение

Спецификация открытого формата виртуализации (OVF) определяет открытый, безопасный, переносимый, эффективный и расширяемый формат для упаковки и распространения программного обеспечения, предназначенного для выполнения на виртуальных машинах. Ключевыми свойствами формата являются следующие:

-Оптимизирован для распространения.— OVF поддерживает проверку контента и контроль целостности на основе инфраструктуры управления стандартными открытыми ключами, что обеспечивает базовую схему управления лицензированием программного обеспечения.

- **Оптимизирован для простого, автоматизированного взаимодействия с пользователем**. — Для управления жизненным циклом процесса OVF поддерживает проверку всего пакета и каждой виртуальной машины или метаданных компонентов OVF в течение фазы установки виртуальной машины (ВМ). Он также предоставляет соответствующий пакет пользовательской описательной информации, которая может быть использована платформой виртуализации для оптимизации процесса установки.

- **Поддерживает конфигурации как одиночных ВМ, так и многих ВМ.** — OVF поддерживает и стандартные структуры из одиночных ВМ, и структуры, обеспечивающие сложное, многоуровневое обслуживание и состоящие из многочисленных взаимозависимых ВМ.

- **Переносимая упаковка для ВМ.** — OVF представляет собой нейтральную платформу виртуализации, которая также допускает использование специфичных для конкретной платформы улучшений. OVF поддерживает полный спектр форматов виртуальных жестких дисков, используемых в настоящее время в программах управления операционными системами (hepervisor), а кроме того, допускает расширение, которое позволяет дополнить спектр, форматами, которые могут возникнуть в будущем. Свойства виртуальной машины фиксируются кратко и точно.

- **Независимость от поставщика и платформы.** — OVF не предполагает использование определенных платформы узла, платформ виртуализации или гостевой операционной системы.

- **Расширяемость.** — помимо того, что OVF готов к использованию, он и расширяем. OVF разработан таким образом, что позволяет расширение, поскольку индустрия технологий виртуальных устройств непрерывно развивается. OVF также поддерживает и допускает кодирование специфичных для поставщика метаданных для поддержки определенных вертикальных рынков.

- **Локализуемость:** OVF допускает включение описаний, предназначенных для прочтения пользователем, в нескольких местах. Таким образом поддерживается локализация интерактивных процессов во время установки. Эта возможность позволяет использовать единственную упаковку многократно для обслуживания различных потребностей рынка.

- **Открытость стандарта:** OVF стал результатом сотрудничества ключевых поставщиков в отрасли, разработан и принят на отраслевом форуме в качестве стандарт будущего для переносимых виртуальных машин.

OVF не призван быть эффективным форматом выполнения. Допускается, но не требуется обязательное использование программы управления операционными системами (hepervisor) для выполнения программного обеспечения на виртуальных машинах непосредственно из открытого формата виртуализации.

Этот стандарт содержит четыре приложения. Приложение С является нормативном и считается частью этого стандарта. Приложения А, В, и D — информативные и не считаются его частью.

Запросы на интерпретацию, предложения по улучшение и применению или отчеты о недостатках приветствуются. Они должны быть отправлены в Международный комитет Стандартов Информационных технологий (INCITS), ITI, 1101 K Street, NW, Suite 610, Washington, DC 20005.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии
СПЕЦИФИКАЦИЯ ОТКРЫТОГО ФОРМАТА ВИЗУАЛИЗАЦИИ (OVF)

Information technology – Open Virtualization Format (OVF) specification

Дата введения – 2015–01–01

1 Область применения

- Спецификация открытого формата виртуализации (OVF) описывает открытый, безопасный, переносимый, эффективный и расширяемый формат для упаковки и распространения программного обеспечения, которое предназначено для выполнения на виртуальных машинах.

2 Нормативные ссылки

Перечисленные далее стандарты содержат положения, которые посредством ссылок в этом тексте составляют положения данного американского национального стандарта. Все стандарты подлежат пересмотру, и сторонам соглашений, основанных на этом американском национальном стандарте, рекомендуется изучить возможность применения самых последних изданий следующих стандартов.

Стандарт ANSI/IEEE 1003.1-2008, Стандарт IEEE для информационных технологий – Спецификация интерфейса переносимых операционных систем (POSIX) Основы, Выпуск 7, Институт инженеров по электронике и радиотехнике, декабрь 2008,

gost.e-ecolog.ru

Схема DMTF CIM 2.22,

gost.e-ecolog.ru

DMTF DSP0004, Спецификация инфраструктуры типовой информационной модели *(CIM) 2.5*,

gost.e-ecolog.ru

DMTF DSP0230,Спецификация отображения *WS-CIM 1.0*,

gost.e-ecolog.ru

DMTF DSP1041, Профиль назначения ресурсов *(RAP) 1.1*,

gost.e-ecolog.ru .pdf

DMTF DSP1043, Профиль возможности размещения *(ACP) 1.0*,

gost.e-ecolog.ru

IETF RFC1738, Тим Бернерс-Ли, Единый указатель ресурсов *(URL)*, декабрь 1994,

gost.e-ecolog.ru 738.txt

IETF RFC1952, П. Дойч, Версия 4.3 спецификации формата файла GZIP, май 1996

gost.e-ecolog.ru 952.txt

IETF RFC5234, Расширенная BNF для спецификаций синтаксиса*: ABNF*,
http://www.ietf.org/rfc/rfc5234.txt

IETF RFC2616, Р. Филдинг и др., Гипертекстовый Протокол передачи – HTTP/1.1, июнь 1999,

gost.e-ecolog.ru

IETF RFC3986, Универсальные идентификаторы ресурса (URI): Общий синтаксис,
http://www.ietf.org/rfc/rfc3986.txt

ISO 9660, 1988 Обработка информации. Структура тома и файлов на компакт-диске (CD-ROM) для обмена информацией,
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=17505

Издание официальное

Директивы ИСО/МЭК. Часть 2.I Правила построения и формулирования международных стандартов

gost.e-ecolog.ru elink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

Примечание – *При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.*

3 Термины и определения

Для целей данного документа применяются следующие термины и определения.

3.1 **Может** (can): Используется для утверждения возможности или способности материальной, физической или причинной.

3.2 **Не может** (cannot) Используется для утверждения возможности или способности, материальной, физической или причинной.

3.3 **Условные** (conditional): Указывает на требования, которые должны строго соответствовать документу, в случае соблюдения указанных условий.

3.4 **Обязательные** (mandatory): Определяют требования, которые должны строго соответствовать документу и отклонения от которых не допускается.

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

3.6 **Не является необходимым (**need not): Указывает на образ действия, допустимый в рамках документа.

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

3.8 **Обязан** (shall): Определяет требования, которые должны строго соответствовать документу и отклонения от которых не допускается.

3.9 **Не обязан** (shall not): Определяет требования, которые должны строго соответствовать документу, и отклонения от которых не допускается.

3.10 **Следует** (should): Указывает, что из нескольких возможностей, одна рекомендуется как наиболее подходящая, не упоминая и не исключая других, или же что определенный образ действия предпочтителен, но не обязателен.

3.11 **Не следует** (should not): Указывает, что определенная возможность или образ действия нежелательны, но не запрещены.

3.12 **Система** (Appliance): См. Виртуальная система.

3.13 **Платформа развертывания** (deployment platform) Программный продукт, который осуществляет установку пакета OVF.

3.14 **Гостевое программное обеспечение** (guest software): Программное обеспечение, хранимое на виртуальных дисках и запускаемое в момент запуска виртуальной машины. Обычно это операционная система и некоторые приложения и службы уровня пользователя.

3.15 **Пакет OVF** (OVF package): Файл XML- описания OVF XML плюс ноль или более файлов.

3.16 **Описание OVF** (OVF descriptor): XML-файл описания OVF.

3.17 **Платформа** (platform): См. Платформа развертывания (3.13).

3.18 **Виртуальная система** (virtual appliance): Сервис, поставляемый в виде полного стека программного обеспечения для установки на одной или более виртуальных машинах. Предполагается, что обычно виртуальные системы будут поставляться в виде пакетов OVF.

3.19 **Виртуальные аппаратные средства** (virtual hardware): Аппаратные средства (включая центральный процессор, контроллеры, устройства Ethernet и диски), которые могут быть использованы гостевым программным обеспечением.

3.20 **Виртуальная машина** (virtual machine): Полная среда, которая поддерживает выполнение гостевого программного обеспечения. Виртуальная машина — это полная инкапсуляция виртуальных аппаратных средств, виртуальных дисков и связанных с ними метаданных. Виртуальные

машины позволяют мультиплексировать возможности базовой физической машины через посредство уровня программного обеспечения, называемого гипервизором.

3.21 **Набор виртуальных машин** (virtual machine collection): Комплекс, состоящий из ряда виртуальных машин. Он может быть простым набором одной или более виртуальных машин, или это может быть сложный комплекс, созданный из комбинации виртуальных машин и других наборов виртуальных машин. Поскольку допускается составление наборов виртуальных машин, то допускается и вложенность компонентов комплекса.

4 Условные обозначения и сокращения

В документе используются условные обозначения и сокращения, перечисленные далее:

4.1.1 **CIM** – Типовая информационная модель (Common Information Model).

4.1.2 **IP** – Протокол интернета (Internet Protocol).

4.1.3 **OVF** – Открытый формат виртуализации (Open Virtualization Format).

4.1.4 **VM** – Виртуальная машина (Virtual Machine).

5 Пакеты OVF

5.1 Структура пакета OVF

Пакет OVF должен содержать в себе следующие файлы:

- один дескриптор OVF с расширением .ovf;
- ноль или один манифест OVF с расширением .mf;
- ноль или один сертификат OVF с расширением .cert;
- ноль или более файлов образа диска;
- ноль или более дополнительных файлов ресурсов, таких как образы дисков ISO;

Необходимо, чтобы были использованы расширения файла.ovf, .mf и .cert.

ПРИМЕР 1 — Следующий список файлов является примером пакета OVF:

package.ovf

package.mf

de-DE-resources.xml

vmdisk1.vmdk

vmdisk2.vmdk

resource.iso

ПРИМЕЧАНИЕ — В приведенном примере используются дисковые файлы VMDK, однако, поддерживаются и многодисковые форматы.

Пакет OVF может быть сохранен или в виде единого модуля, или как набор файлов, как описано в 5.3 и 5.4. Должны поддерживаться оба эти режима.

В пакете OVF может быть файл манифеста, содержащий хэш-суммы SHA-1 отдельных файлов в пакете. Файл манифеста должен иметь расширение .mf, то же самое базовое имя, что и файл .ovf, и быть одноуровневым элементом файла .ovf. Если присутствует файл манифеста, то потребитель пакета OVF должен проверить хэш-суммы, вычисляя фактические хэш-суммы SHA-1 и сравнивая их с суммами, представленными в файле манифеста.

Определения синтаксиса далее используют ABNF с исключениями, перечисленными в приложении А.

Формат файла манифеста следующий:

```
manifest_file = *( file_digest )
file_digest = algorithm "(" file_name ")" "=" sp digest nl
algorithm = "SHA1"
digest = 40( hex-digit ) ; 160-bit digest in 40-digit hexadecimal
hex-digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "a" |
"b" | "c" | "d" | "e" | "f"
sp = %x20
nl = %x0A
```

ПРИМЕР 2 — В примере показано частичное содержание файла манифеста:

SHA1(package.ovf)= 237de026fb285b85528901da058475e56034da95

SHA1(vmdisk1.vmdk)= 393a66df214e192ffbfedb78528b5be75cc9e1c3

Пакет OVF может иметь подпись посредством подписи файла манифеста. Сумма файла манифеста размещается в файле сертификата с расширением .cert вместе с закодированным base64 сертификатом X.509. Файл .cert должен иметь то же самое базовое имя, что и файл .ovf, и быть одноуровневым элементом файла ovf. Потребитель пакета OVF должен проверить подпись и подлинность сертификата. Формат файла сертификата должен быть следующим:

certificate_file = manifest_digest certificate_part
manifest_digest = algorithm "(" file_name ")" "=" sp signed_digest nl
algorithm = "SHA1"
signed_digest = *(hex-digit)
certificate_part = certificate_header certificate_body certificate_footer
certificate_header = "-----BEGIN CERTIFICATE-----" nl
certificate_footer = "-----END CERTIFICATE-----" nl
certificate_body = base64-encoded-certificate nl
; base64-encoded-certificate is a base64-encoded X.509
; certificate, which may be split across multiple lines
hex-digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "a"
| "b" | "c" | "d" | "e" | "f"
sp = %x20
nl = %x0A

ПРИМЕР 3 — Приведенный список файлов является примером пакета OVF с подписью:

package.ovf
package.mf
package.cert
de-DE-resources.xml
vmdisk1.vmdk
vmdisk2.vmdk
resource.iso

ПРИМЕР 4 — Следующий пример отражает содержание файла сертификата OVF, в котором сумма SHA1 файла манифеста была подписана с ключом длиной 512 битов:

SHA1(package.mf)= 7f4b8efb8fe20c06df1db68281a63f1b088e19dbf00e5af9db5e8e3e319de
7019db88a3bc699bab6ccd9e09171e21e88ee20b5255cec3fc28350613b2c529089
-----BEGIN CERTIFICATE-----
MIIBgjCCASwCAQQwDQYJKoZIhvcNAQEEBQAwODELMAkGA1UEBhMCQVUxDDAKBgNV
BAgTA1FMRDEbMBkGA1UEAxMSU1NMZWF5L3JzYSB0ZXN0IENBMB4XDTk1MTAwOTIz
MzIwNVoXDTk4MDcwNTIzMzIwNVowYDELMAkGA1UEBhMCQVUxDDAKBgNVBAgTA1FM
RDEZMBcGA1UEChMQTWIuY29tIFB0eS4gTHRkLjELMAkGA1UECxMCQ1MxGzAZBgNV
BAMTEINTTGVheSBkZW1vIHNlcnZlcjBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQC3
LCXcScWua0PFLkHBLm2VejqpA1F4RQ8q0VjRiPafjx/Z/aWH3ipdMVvuJGa/wFXb
/nDFLDIfWp+oCPwhBtVPAgMBAAEwDQYJKoZIhvcNAQEEBQADQQArNFsihWljBzb0
DcsU0BvL2bvSwJrPEqFlkDq3F4M6EgutL9axEcANWgbbEdAvNJD1dmEmoWny27Pn
lms6ZOZB
-----END CERTIFICATE-----

Файлы манифеста и сертификата в случае их наличия не должны включаться в раздел ссылок (References) дескриптора OVF (см. 7.1). Этим гарантируется независимость содержимого

дескриптора OVF от того, имеет ли пакет OVF манифест и подписан ли он. Поэтому решение добавить в пакет манифест или сертификат может быть принято на более позднем этапе.

Расширения файла .mf и .cert могут использоваться для других файлов в пакете OVF при условии, что они не находятся на том уровне URL или пути доступа, где они могли бы быть интерпретированы как манифест или сертификат пакета.

5.2 Виртуальные дисковые форматы

OVF не требует использования определенного дискового формата, но для определения его спецификации должен присутствовать унифицированный идентификатор ресурса URI, который идентифицирует необходимую для интерпретации дискового формата спецификацию. Спецификация не должна быть машиночитаемой, но она должна быть статична и уникальна, чтобы URI мог использоваться программным обеспечением в процессе чтения пакета OVF в качестве ключа для однозначного определения формата диска. Спецификация должна предоставить информацию, достаточную для того, чтобы квалифицированный человек мог должным образом интерпретировать дисковый формат как для чтения, так и для записи дисковых данных. Рекомендуется, чтобы эти URI были разрешимы.

5.3 Распространение в виде одного файла

Пакет OVF может быть записан как единственный файл, используя формат TAR. Расширение того файла должно быть .ova (открытая виртуальная система или приложение).

ПРИМЕР — Следующий пример показывает образец имени файла для пакета OVF этого типа:

D:\virtualappliances\myapp.ova

Для пакетов OVF, сохраненных в виде одного файла, все ссылки на файлы в дескрипторе OVF должны быть ссылками относительного пути и должны указать на файлы, включенные в архив TAR. Относительные директории в архиве допускаются, однако, ссылки относительного пути не должны содержать точечных сегментов «..». Обычно инструмент извлечения файлов TAR должен просмотреть весь архив, даже в случае, если требуемый файл уже найден вначале, потому что в конец архива могут быть добавлены заменяющие файлы без изменения начальной части архива. Для файлов OVF TAR дублирование в пределах архива не допускается. Кроме того, в архиве файлы должны быть в следующем порядке:

1) Дескриптор OVF.
2) Манифест OVF (не обязательно).
3) Сертификат OVF(не обязательно).
4) Остальные файлы должны быть в том же самом порядке, как перечислено в разделе ссылок (References) (см. 7.1). Необходимо отметить, что все файлы наборов внешних ресурсов обработки строк для интернационализации должны быть первыми в разделе ссылок (см. раздел 10).
5) Манифест OVF (не обязательно).
6) Сертификат OVF(не обязательно).

Важно, что файл сертификата является дополнительным. Если файл сертификата отсутствует, то файл манифеста тоже является дополнительным. Если файлы манифеста или сертификата присутствуют, то оба они должны либо размещаться после дескриптора OVF, либо оба должны быть размещены в конце архива.

При развертывании ограничения упорядочивания гарантируют возможность извлечения дескриптора OVF из файла OVF TAR без сканирования всего архива. При генерации ограничения упорядочивания гарантируют, что файл OVF TAR может легко использоваться для генерации «на лету». Ограничения не препятствуют созданию файлов OVF TAR с использованием стандартных инструментов упаковки TAR.

Используемый формат TAR должен соответствовать формату USTAR (Универсальный Стандартный Ленточный Архив), как он определен группой стандартов POSIX IEEE 1003.1.

5.4 Распространение в виде нескольких файлов

Пакет OVF может быть сделан доступным в виде набора файлов, например, на стандартном веб-сервере.

***ПРИМЕР — Пример пакета OVF в виде набора файлов на веб-сервере*:**

gost.e-ecolog.ru

gost.e-ecolog.ru **.vmdk**

gost.e-ecolog.ru

gost.e-ecolog.ru

gost.e-ecolog.ru

6 Дескриптор OVF

Все метаданные о пакете и его содержании хранятся в дескрипторе OVF. Это расширяемый XML-документ, предназначенный для закодированной информации, такой как особенности продукта, требования к виртуальным аппаратным средствам и к лицензированию.

XML-схема файла определения dsp8023_1.1.0.xsd для дескриптора OVF содержит элементы и атрибуты.

Разделы 7, 8, и 9 описывают семантику, структуру и основы расширяемости дескриптора OVF.

Эти разделы не предназначены для того, чтобы заменить собой определения схемы, а дополняют эти определения.

XML-документ дескриптора OVF должен содержать один элемент конверта (Envelope), который является единственным элементом, допускаемым на верхнем уровне.

Пространства имен XML, используемые в этой спецификации, перечислены в Таблице 1. Выбор любого префикса пространства имен является произвольным и не выступает семантически существенным.

Таблица 1 – Префиксы пространства имен XML

Префикс	Пространство имен XML
ovf	gost.e-ecolog.ru
ovfenv	gost.e-ecolog.ru
rasd	gost.e-ecolog.ru
vssd	gost.e-ecolog.ru
cim	gost.e-ecolog.ru

7 Элемент конверт (Envelope)

Элемент «конверт» (Envelope) описывает все метаданные для виртуальных машин (включая виртуальные аппаратные средства), а кроме того − и структуру непосредственно пакета OVF.

Самый внешний уровень конверта состоит из следующих частей:

- указание версии, определенной унифицированным указателем ресурса URI пространства имен XML;
- список файловых ссылок на все внешние файлы, которые являются частью пакета OVF, определенного элементом «ссылки» (References) и его дочерним элементом «файл»(File). Обычно файлы − это виртуальные дисковые файлы, образы дисков ISO и ресурсы интернационализации;
- часть метаданных, определенная элементами раздела, как определено в разделе 9.
- описание контента либо одна виртуальная машина (элемент VirtualSystem), либо набор нескольких виртуальных машин (элемент VirtualSystemCollection);
- спецификация набора ресурсов обработки сообщений для нуля или большего числа локалей, определенная посредством элемента «строка» (Strings) для каждой локали.

ПРИМЕР — В примере показана структура дескриптора OVF с элементом «конверт» (Envelope) верхнего уровня

<?xml version="1.0" encoding="UTF-8"?>

<Envelope xmlns:xsi="gost.e-ecolog.ru **"**

xmlns:vssd="gost.e-ecolog.ru

gost.e-ecolog.ru **"**

xmlns:rasd="gost.e-ecolog.ru

schema/2/CIM_ResourceAllocationSettingData"

xmlns:ovf="gost.e-ecolog.ru **"**

xmlns="gost.e-ecolog.ru **"**

```
            xml:lang="en-US">
            <References>
              <File ovf:id="de-DE-resources.xml" ovf:size="152 40"
                    ovf:href="gost.e-ecolog.ru                                    >
              <File ovf:id="file1" ovf:href="vmdisk1.vmdk" ovf:size="180114671"/>
              <File ovf:id="file2"   ovf:href="vmdisk2.vmdk" ovf:size="4882023564"
ovf:chunkSize="2147483648"/>
              <File ovf:id="file3" ovf:href="resource.iso" ovf:size="212148764"
ovf:compression="gzip"/>
              <File ovf:id="icon" ovf:href="icon.png" ovf:size="1360"/>
            </References>
            <!-- Приводится метаинформация о всех виртуальных дисках пакета -- >
            <DiskSection>
                  <Info> Описан набор виртуальных дисков</Info>
                  <!-- Дополнительный раздел контента-->
             </DiskSection>
            <!-- Описаны все сети, используемые пакетом-- >
             <NetworkSection>
                  <Info> Список логических сетей, используемых пакетом </Info>
                  <!-- Дополнительный раздел контента -->
            </NetworkSection>
            <SomeSection ovf:required="false">
                  <Info> Описание контента в виде обычного текста </Info>
                  <!-- Дополнительный раздел контента -- >
            </SomeSection>
            <!-- Additional sections can follow -- >
            <VirtualSystemCollection ovf:id="Some Product">
                  <!-- Дополнительные разделы, включая VirtualSystem или
VirtualSystemCollection– >
            </VirtualSystemCollection >
             <Strings xml:lang="de-DE">
             <!-- Спецификация набора ресурсов обработки  сообщений для локали de-DE -- >
            </Strings>
      </Envelope>
```

Дополнительный атрибут xml:lang в элементе «конверт» (Envelope) должен определять для сообщений в дескрипторе локаль по умолчанию. Дополнительные элементы «строки» (Strings) должны содержать наборы строковых ресурсов для различных локалей. Более детально поддержка интернационализации рассматривается в разделе 10.

7.1 Файловые ссылки (References)

Раздел файловых ссылок, определенный посредством элемента «ссылки» (References), позволяет программе легко проверить целостность пакета OVF без необходимости анализировать или интерпретировать всю структуру дескриптора. Программные средства могут безопасно манипулировать (например, копировать или архивировать) пакетами OVF без риска потери файлов. Файлы внешних наборов строковых ресурсов для интернационализации должны быть помещены в начало в элемента «ссылки» (References). (См. раздел 10).

Каждому элементу File в разделе ссылок необходимо, используя атрибут ovf:id, приписать идентификатор. Идентификатор должен быть уникальным в пакете OVF. Каждый элемент File должен быть определен с использованием атрибута ovf:href, в котором должен содержаться URL. Должны поддерживаться как схема ссылок относительного пути, так и схемы URL: «file», «http», и «https»,см. RFC1738 и RFC3986. Другие схемы URL не должны использоваться. Если не определена никакая

схема URL, то значение атрибута ovf:href должно интерпретироваться как путь доступа к файлу, на который указывает ссылка, непосредственно относительно расположения дескриптора OVF. Относительный путь доступа должен использовать синтаксис ссылок относительного пути в RFC3986. Файл, на который указывает ссылка, должен существовать. Два различных элемента File своими атрибутами ovf:href не должны ссылаться на один и тот же файл.

Каждый файл, на который ссылается элемент File, может быть сжат с использованием gzip (см. RFC1952). Если элемент File будет сжат, используя gzip, то атрибут ovf:compression должен быть установлен в «gzip». В противном случае значение атрибута ovf:compression должен быть «identity» или же атрибут должен быть опущен полностью. Альтернативно, если ссылка href — это HTTP или HTTPS URL, то сжатие может быть определено сервером HTTP при использовании Content-Encoding: gzip в заголовке http (см. RFC2616). Допускается использование кодирование контента HTTP в комбинации с атрибутом ovf:compression, однако в общем случае степень сжатия при этом не улучшается. При использовании сжатия атрибут ovf:size должен определять фактический размер сжатого файла.

Файлы, на которые указывают ссылки в разделе ссылок, могут быть разбиты на части для того, чтобы обойти ограничения на размеры файлов в определенных файловых системах. Разбиение на блоки должно быть обозначено присутствием атрибута ovf:chunkSize. Значение ovf:chunkSize должно быть размером каждого блока, кроме последнего, размер которого может быть меньше.

Если определен ovf:chunkSize, то элемент File должен указывать на файл блока, представляющий блок всего файла. В этом случае значение атрибута ovf:href определяет только часть URL, и синтаксис URL для ссылки на файл блока показан далее. Синтаксис использует ABNF с исключениями, перечисленными в приложении A.

chunk-url = href-value "." chunk-number

chunk-number = 9(decimal-digit)

decimal-digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

В этом примере значение href − это значение атрибута ovf:href, а номер блока − это положение блока относительно блока 0. Нумерация блоков начинается с значения 0 и увеличивается на 1 для каждого последующего блока.

Разделение на блоки может быть объединено со сжатием. В этом случае весь файл должен быть сжат перед разбиением на блоки, и сжатый файл должен быть разделен на равные части сжатого файла, за исключением последнего блока, размер которого может быть меньше.

Если в пакете OVF имеется файл манифеста, то имя файла в записях манифеста должно соответствовать значению атрибута ovf:href для этого файла, за исключением случая, если файл разбивается на множество частей, когда должно использоваться chunkurl, и в файл манифеста должны входить записи для каждого отдельного блока. Для разделенных на блоки файлов допускается, чтобы файл манифеста содержал запись для всего файла. В таком случае должна также быть проверена соответствующая хэш-сумма.

ПРИМЕР 1 — В следующем примере показаны различные типы ссылок на файл:

<File ovf:id="disk1" ovf:href="disk1.vmdk"/>

<File ovf:id="disk2" ovf:href="disk2.vmdk" ovf:size="5368709120"

ovf:chunkSize="2147483648"/>

<File ovf:id="iso1" ovf:href="resources/image1.iso"/>

<File ovf:id="iso2" ovf:href=" gost.e-ecolog.ru **>**

Пример 2 — В следующем примере показаны записи манифеста, соответствующие файловым ссылкам из предыдущего примера :

SHA1(disk1.vmdk)= 3e19644ec2e806f38951789c76f43e4a0ec7e233

SHA1(disk2.vmdk.000000000)= 4f7158731ff434380bf217da248d47a2478e79d8

SHA1(disk2.vmdk.000000001)= 12849daeeaf43e7a89550384d26bd437bb8defaf

SHA1(disk2.vmdk.000000002)= 4cdd21424bd9eeafa4c42112876217de2ee5556d

SHA1(resources/image1.iso)= 72b37ff3fdd09f2a93f1b8395654649b6d06b5b3

SHA1(http://mywebsite/resources/image2.iso)=d3c2d179011c970615c5cf10b30957d1c4c968ad

7.2 Элемент контента Content

Конфигурация виртуальной машины в пакете OVF представлена элементом VirtualSystem или

VirtualSystemCollection. Этим элементам необходимо дать идентификатор, используя атрибут ovf:id. Прямые дочерние элементы VirtualSystemCollection должны иметь уникальные идентификаторы.

В схеме OVF элементы VirtualSystem и VirtualSystemCollection являются частью группы замены с элементом Content в качестве главного элемента группы замены. Элемент Content абстрактен и не может быть использован напрямую. В дескрипторе OVF должны быть один или более элементов Content.

Элемент VirtualSystem описывает единственную виртуальную машину и является просто контейнером элементов раздела. Элементы этого раздела описывают виртуальные аппаратные средства, ресурсы, и информацию о продукте и подробно рассмотрены в разделах 8 и 9.

Структура элемента VirtualSystem следующая:

```
<VirtualSystem ovf:id="simple-app">
        <Info>A virtual machine</Info>
        <Name>Simple Appliance</Name>
        <SomeSection>
<!-- Дополнительный раздел контента -->
        </SomeSection>
        <!—Возможны дополнительные разделы -->
</VirtualSystem>
```

Элемент VirtualSystemCollection − это контейнер нескольких элементов VirtualSystem или VirtualSystemCollection. Таким образом могут быть описаны произвольные сложные конфигурации. Элементы раздела на уровне VirtualSystemCollection содержат информацию о системе, свойствах, требованиях ресурсов и прочем, что подробно рассмотрено в разделе 9.

Структура элемента VirtualSystemCollection следующая:

```
<VirtualSystemCollection ovf:id="multi-tier-app">
  <Info>A collection of virtual machines</Info>
  <Name>Multi-tiered Appliance</Name>
  <SomeSection>
<!-- Дополнительный раздел контента -->
  </SomeSection>
  <!-- Возможны дополнительные разделы -->
  <VirtualSystem ovf:id="...">
<!-- Дополнительные разделы ->
  </VirtualSystem>
<!-- Возможны дополнительные эелементы VirtualSystem или VirtualSystemCollection --
></VirtualSystemCollection>
```

Все элементы в группе замены Content должны содержать элемент Info и могут содержать элемент названия Name. Элемент Info содержит удобочитаемое описание значения этого объекта. Элемент Name это дополнительное локализуемое отображаемое название контента. Подробности локализации элементов Info и Name приводятся в разделе 10.

7.3 Возможности расширения

Данная спецификация позволяет добавлять в дескрипторы OVF пользовательские метаданные несколькими способами.

- Новые элементы раздела могут быть определены как часть группы замены Section и использоваться там, где схемы OVF допускают присутствие разделов (Sections). Все подтипы Section имеют элемент Info, который содержит удобочитаемое описание значения этого объекта. Значения элементов Info могут использоваться, например, для выдачи важных предупреждений пользователям в том случае, если раздел пропускается, даже если синтаксический анализатор ничего не знает о разделе. Детали локализации элемента Info рассмотрены в разделе 10.

- Схемы OVF используют открытую модель контента, в которой ко всем существующим типам в конце могут быть добавлены дополнительные элементы. Точки расширения представлены в схемах OVF объявлениями xs:any с пространством имен = «##other».

- Схемы OVF допускают использование дополнительных атрибутов существующих типов.

Пользовательские расширения не должны использовать пространства имен XML, определенные в этой спецификации. Это касается как пользовательских элементов, так и пользовательских атрибутов.

Значение атрибута ovf:required (Булевская переменная) в пользовательских элементах определяет, является ли информация элемента необходимой для корректного выполнения или служит дополнительной. Если не определено явно, то ovf:required приписывается значения по

умолчанию значение ИСТИНА (TRUE). Потребитель пакета OVF, который обнаруживает требующееся расширение, которое он не понимает, должен остановить выполнение.

Если обнаружены дополнительные дочерние элементы известных элементов Section, которые непоняты и значение их атрибута ovf:required ИСТИНА (TRUE), то потребитель пакета OVF должен интерпретировать весь раздел как непонятный. Такая проверка не является рекурсивной; это применимо только к прямым дочерним элементам элемента Section.

Подобные функциональные возможности гарантируют, что более старые синтаксические анализаторы отклонят более новые спецификации OVF в случаях, если им явно не указано не делать это.

Информация пользовательских атрибутов не должна быть необходимой для корректного функционирования.

Пример 1 —

```
<!—- Пример дополнительного раздела пользователя -->
 <otherns:IncidentTrackingSection ovf:required="false">
 <Info> Определяет информацию, полезную при отслеживании инцидентов </Info>
<BuildSystem>Acme Corporation Official Build System</BuildSystem>
<BuildNumber>102876</BuildNumber>
<BuildDate>10-10-2008</BuildDate>
 </otherns:IncidentTrackingSection>
```

Пример 2 —

```
<!—- Пример открытого контента (расширение существующего типа) -->
<AnnotationSection>
 <Info> Определяет краткое описание этой виртуальной машины </Info>
 <Annotation> Это-пример того, как новый элемент (Author) все же может быть проанализирован старыми клиентами
</Annotation>
 <!—Раздел AnnotationSection расширен элементом Author-->
 <otherns:Author ovf:required="false">John Smith</otherns:Author>
</AnnotationSection>
```

Пример 3 —

```
<!—- Пример дополнительного атрибута пользователя -->
 <Network ovf:name="VM network" otherns:desiredCapacity="1 Gbit/s">
<Description>Основная сеть для виртуальных машин </Description>
 </Network>
```

7.4 Соответствие

Эта спецификация определяет три уровня соответствия для дескрипторов OVF. Высшим уровнем соответствия является 1:

- В дескрипторе OVF используются только разделы и элементы и атрибуты, которые определены в данной спецификации.

Уровень соответствия: 1.

- В дескрипторе OVF используются пользовательские разделы, элементы или атрибуты, которые не определены в этой спецификации, но все такие расширения являются дополнительными в соответствии с определениями 7.3.

Уровень соответствия: 2.

- В дескрипторе OVF используются пользовательские разделы или элементы, которые не определены данной спецификацией, и требуется, как минимум, одно такое расширение, как определено в 7.3. Определения всех необходимых расширений должны быть публично доступны через открытую и полноценную схему XML. Полная спецификация может либо содержаться в XML-схеме, либо быть доступной в виде отдельного документа.

Уровень соответствия: 3.

Уровень соответствия 3 ограничивает переносимость, и его использования следует избегать там, где это возможно.

8 Описание виртуального аппаратного обеспечения

8.1 Раздел виртуальных аппаратных средств VirtualHardwareSection

Каждый элемент VirtualSystem может содержать один или более элементов VirtualHardwareSection, каждый из которых описывает виртуальные аппаратные средства, требуемые виртуальной системой. Виртуальные аппаратные средства, требуемые виртуальной машиной, определены в элементах VirtualHardwareSection. Данная спецификация поддерживает абстрактные или неполные описания аппаратных средств, в которых описаны только главные устройства. Допускается создание гипервизором дополнительных виртуальных аппаратных контроллеров и устройств при условии, что необходимые устройства, перечисленные в дескрипторе, реализованы.

Такое описание виртуальных аппаратных средств основано на классах CIM_VirtualSystemSettingData и CIM_ResourceAllocationSettingData модели CIM. Представление XML модели CIM основано на отображении WS-CIM(DSP0230).

ПРИМЕР — Пример VirtualHardwareSection

```
<VirtualHardwareSection ovf:id="minimal" ovf:transport="iso">
  <Info>500Mb, 1 CPU, 1 disk, 1 nic virtual machine</Info>
  <System>
    <vssd:ElementName>Virtual System Type</vssd:ElementName>
    <vssd:InstanceID>0</vssd:InstanceID>
    <vssd:VirtualSystemType>vmx-4</vssd:VirtualSystemType>
  </System>
  <Item>
    <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
    <rasd:Description>Memory Size</rasd:Description>
    <rasd:ElementName>512 MB of memory</rasd:ElementName>
    <rasd:InstanceID>2</rasd:InstanceID>
    <rasd:ResourceType>4</rasd:ResourceType>
    <rasd:VirtualQuantity>512</rasd:VirtualQuantity>
  </Item>
  <!—Возможны дополнительные элементы Item -->
</VirtualHardwareSection>
```

Элемент VirtualSystem должен иметь прямой дочерний элемент VirtualHardwareSection. Элемент VirtualHardwareSection как прямой дочерний элемент элемента VirtualSystemCollection и элемента Envelope недопустим.

В пределах одного элемента VirtualSystem допускается присутствие нескольких элементов VirtualHardwareSection. Потребитель пакета OVF должен выбрать максимально соответствующее описание виртуального аппаратного обеспечения для конкретной платформы виртуализации. Элемент VirtualHardwareSection может содержать атрибут ovf:id, используемый для идентификации элемента. В случае его наличия значение этого атрибута должно быть уникальным в пределах VirtualSystem.

Атрибут ovf:transport определяет типы транспортных механизмов, посредством которых свойства передаются на виртуальную машину в документе среды OVF. Этот атрибут поддерживает настраиваемую и расширяемую архитектуру, для того, чтобы обеспечить коммуникационные механизмы гостя/платформы. Может быть определено несколько типов транспорта, разделенных одиночным пробелом. Описания свойств приведены в 9.5, а описания типов транспорта и сред OVF — в разделе 11.

Элемент vssd:VirtualSystemType определяет идентификатор типа виртуальной системы, являющийся строкой определения реализации, которая однозначно определяет тип виртуальной системы. Например, для четвертого поколения виртуальных аппаратных средств VMware идентификатор типа виртуальной системы может быть vmx-4, а для третьего поколения виртуальных аппаратных средств Xen — xen-3. Допускается ноль или более идентификаторов типа виртуальной системы, разделенные одним символом пробела. Для виртуальной системы OVF, чтобы быть развертываемой на целевой платформе, виртуальная машина на целевой платформе должна

поддерживать по крайней мере один из типов виртуальной системы, идентифицированных в элементах vssd:VirtualSystemType. Идентификаторы типа виртуальной системы, определенные в элементах vssd:VirtualSystemType, как ожидают, будут противопоставляться значениям свойств VirtualSystemTypesSupported класса CIM_VirtualSystemManagementCapabilities модели CIM.

Характеристики виртуального аппаратного обеспечения описаны как последовательность элементов Item. Элемент Item — это представление XML экземпляра класса CIM_ResourceAllocationSettingData модели CIM. Элемент может описать все требования к памяти и центральному процессору, так же как и требования к виртуальным устройствам.

В элементе Item могут быть определены несколько подтипов устройства, разделенные одним символом пробела.

ПРИМЕР

<rasd:ResourceSubType>buslogic lsilogic</rasd:ResourceSubType>

8.2 Возможности расширения

Дополнительный атрибут ovf:required элемента Item определяет, требуется ли для корректного функционирования гостевого программного обеспечения реализация этого элемента (например, CD-ROM или контроллер USB). Если значение атрибута ovf:required не определено, то его значение по умолчанию ИСТИНА (TRUE).

В дочерних элементах элемента Item дополнительный Булевский атрибут ovf:required должен учитываться, несмотря на то, что эти элементы находятся в разных пространствах имен RASD WS-CIM. Программа, анализирующая элемент Item, должна действовать согласно Таблице 2.

Таблица 2 — Действия для дочерних элементов с атрибутом ovf:required

Дочерний элемент	Значение атрибута ovf:required	Действие
Известен	ИСТИНА (TRUE) или не определен	Нужно интерпретировать элемент Item
Известен	ЛОЖЬ (FALSE)	То же
Неизвестен	ИСТИНА (TRUE) или не определен	Обработка Item должна быть прекращена
Неизвестен	ЛОЖЬ (FALSE)	Элемент Item нужно игнорировать

8.3 Элементы виртуального аппаратного обеспечения

Общая форма любого элемента Item в элементе VirtualHardwareSection имеет следующий вид:

```
<Item ovf:required="..." ovf:configuration="..." ovf:bound="...">
    <rasd:Address> ... </rasd:Address>
    <rasd:AddressOnParent> ... </rasd:AddressOnParent>
    <rasd:AllocationUnits> ... </rasd:AllocationUnits>
    <rasd:AutomaticAllocation> ... </rasd:AutomaticAllocation>
    <rasd:AutomaticDeallocation> ... </rasd:AutomaticDeallocation>
    <rasd:Caption> ... </rasd:Caption>
    <rasd:Connection> ... </rasd:Connection>
    <!—возможны описания нескольких элементов connection -->
    <rasd:ConsumerVisibility> ... </rasd:ConsumerVisibility>
    <rasd:Description> ... </rasd:Description>
    <rasd:ElementName> ... </rasd:ElementName>
    <rasd:HostResource> ... </rasd:HostResource>
    <rasd:InstanceID> ... </rasd:InstanceID>
    <rasd:Limit> ... </rasd:Limit>
    <rasd:MappingBehavior> ... </rasd:MappingBehavior>
    <rasd:OtherResourceType> ... </rasd:OtherResourceType>
```

<rasd:Parent> ... </rasd:Parent>
<rasd:PoolID> ... </rasd:PoolID>
<rasd:Reservation> ... </rasd:Reservation>
<rasd:ResourceSubType> ... </rasd:ResourceSubType>
<rasd:ResourceType> ... </rasd:ResourceType>
<rasd:VirtualQuantity> ... </rasd:VirtualQuantity>
<rasd:Weight> ... </rasd:Weight>
</Item>

Элементы соответствуют свойствам, представленным классом CIM_ResourceAllocationSettingData. Они имеют семантику определенных настроек, как это определено в DSP1041, в любых профилях, полученных из DSP1041 для конкретных типов ресурса, и в настоящем документе.

ПРИМЕР — В следующем примере показано описание объема памяти:

<Item>
<rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
<rasd:Description>Memory Size</rasd:Description>
<rasd:ElementName>256 MB of memory</rasd:ElementName>
<rasd:InstanceID>2</rasd:InstanceID>
<rasd:ResourceType>4</rasd:ResourceType>
<rasd:VirtualQuantity>256</rasd:VirtualQuantity>
</Item>

Элемент описания Description используется, чтобы предоставить дополнительные метаданные об элементе непосредственно. Этот элемент позволяет обеспечить потребителя пакета OVF дескриптивной информацией обо всех элементах, включая и те элементы, которые были неизвестны на момент записи приложения.

Элементы заголовка Caption, описания Description и названия элемента ElementName локализуемы посредством использования атрибута ovf:msgid из пространства имен конверта OVF. Поддержка интернационализации подробно описана в разделе 10.

Дополнительный атрибут ovf:configuration содержит список имен конфигурации. Семантика этого атрибута описана в вариантах развертывания 9.8. Для определения диапазонов используется дополнительный атрибут ovf:bound (см. 8.4).

Такие устройства как диски, CD-ROM, а также сетевые нуждаются в поддержке с стороны платформы развертывания. Требования к поддержке определены с использованием либо элемента HostResource, либо элемента Connection.

Для адаптера Ethernet логическое сетевое имя определено в элементе Connection. Адаптеры Ethernet, которые ссылаются на одно и то же логическое сетевое имя в рамках пакета OVF, должны быть установлены в той же сети.

Элемент HostResource используется для того, чтобы ссылаться на ресурсы, включенные в дескриптор OVF, так же как и логические устройства на платформе развертывания. Значения для элементов HostResource, ссылающихся на ресурсы, включенные в дескриптор OVF, имеют формат унифицированных локаторов ресурсов URI, как это показано в Таблице 3.

Таблица 3 - Элемент HostResource

Содержимое	Описание
ovf:/file/<id>	Ссылка на файл в OVF, как определено в разделе ссылок References. <id> должно быть значением атрибута ovf:id элемента File, на который указывает ссылка
ovf:/disk/<id>	Ссылка на виртуальный диск, как определено в разделе DiskSection. <id> должно быть значением атрибута ovf:diskId элемента diskId, на который указывает ссылка

Если для устройства, которое требует поддержки, никакая поддержка не определена, то платформа развертывания должна сделать соответствующий выбор, например, запрашивая пользователя. Определение более чем одной поддержки для устройства не допускается.

В таблице 4 представлен краткий обзор того, как используются элементы для описания виртуальных устройств и контроллеров.

Таблица 4 – Элементы для виртуальных устройств и контроллеров

Элемент	Описание использования
rasd:Description	Удобочитаемое описание смысла информации. Например, «Определяет объем памяти виртуальной машины»
rasd:ElementName	Удобочитаемое описание контента. Например, «Память 256 МБ»
rasd:InstanceID	Уникальный идентификатор ID экземпляра элемента в пределах раздела
rasd:HostResource	Абстрактно определяет, как устройство должно соединиться с ресурсом на платформе развертывания. Не все устройства нуждаются в поддержке. См. Таблицу 3
rasd:ResourceType rasd:OtherResourceType rasd:ResourceSubtype	Определяет вид (тип) описываемого устройства
rasd:AutomaticAllocation	Для устройств, которые являются присоединяемыми, таких как дискеты, CD-ROM и адаптеры Ethernet, этот элемент определяет, должно ли устройство быть присоединено во включенном состоянии
rasd:Parent	InstanceID родительского контроллера (если такой имеется)
rasd:Connection	Для адаптера Ethernet этот элемент определяет абстрактное сетевое имя соединения для виртуальной машины. Все адаптеры Ethernet, которые определяют то же самое абстрактное сетевое имя соединения в пределах пакета OVF, должны располагаться в одной и той же сети. Абстрактное сетевое имя соединения должно присутствовать в разделе NetworkSection на внешнем уровне конверта
rasd:Address	Определенное устройство. Для адаптера Ethernet этот элемент определяет Мак – адрес
rasd:AddressOnParent	Для устройства этот элемент определяет его положение на контроллере
rasd:AllocationUnits	Определяет единицы выделения для использования. Например, «байт * 2^20»
rasd:VirtualQuantity	Определяет количество предоставленных ресурсов. Например, «256»
rasd:Reservation	Определяет минимальное количество ресурсов, которое гарантировано доступны
rasd:Limit	Определяет максимальное количество предоставляемых ресурсов
rasd:Weight	Определяет относительный приоритет для этого выделения ресурсов относительно других выделений

Здесь перечислены только поля, непосредственно связанные с описанием устройств. Для того чтобы получить полное описания всех полей, следует обратиться к CIM MOF с учетом того, что каждое поле соответствует тождественно именованному свойству в классе CIM_ResourceAllocationSettingData.

8.4 Диапазоны элементов

Дополнительный атрибут ovf:bound может использоваться для того чтобы определить диапазоны для элементов Item. У диапазона есть минимальное, номинальное и максимальное значение, обозначаемые как min, normal и max, где min ≤ normal ≤ max. Значением по умолчанию для min и max является значение, специфицированное для normal.

Платформу, на которой развертывается пакет OVF, рекомендуется начать с номинального значения, а затем в процессе скорректировать значение в пределах диапазона для настройки производительности и проверки допустимости.

Для элементов Item в элементах VirtualHardwareSection и ResourceAllocationSection определена следующая дополнительная семантика.

- Каждый элемент Item имеет дополнительный атрибут ovf:bound. Это значение может быть специфицировано как min, max или normal. Значение по умолчанию — normal. Если же атрибут не определен или определен как normal, то элемент рассматривается, как являющийся частью обычного описания виртуальных аппаратных средств или распределения ресурсов.
- Если значение ovf:bound определено или как min, или как max, то элемент используется, чтобы определить верхнюю или нижнюю границу для одного или более значений для данного InstanceID. Такой элемент вызывают маркером диапазона.

Семантика маркеров диапазона следующая.

- Необходимо, чтобы были определены InstanceID и ResourceType, а ResourceType должен соответствовать другим элементам Item с тем же самым InstanceID.
- Определение больше одного маркера диапазона min или больше одного маркера диапазона max для данного раздела RASD, идентифицированного посредством InstanceID, недопустимо.
- Элемент Item с маркером диапазона должен иметь соответствующий элемент Item без маркера диапазона, т.е. элемент Item — или без атрибута ovf:bound, или с атрибутом ovf:bound, значение которого номинальное. Этот соответствующий элемент определяет значение по умолчанию.
- В случае, если для элемента Item определен только маркер диапазона min, значение max в пределах набора допустимых значений для свойства не ограниченно вверх.
- Если для элемента Item определен только маркер диапазона max, значение min в пределах набора допустимых значений для свойства не ограниченно вниз.
- Значение по умолчанию должно быть в пределах диапазона.
- Использование элементов нецелого типа для маркера диапазона RASD недопустимо.

***ПРИМЕР — Следующий пример показывает использование маркеров* диапазона**

```
<VirtualHardwareSection>
 <Info>...</Info>
  <Item>
      <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
      <rasd:ElementName>512 MB memory size</rasd:ElementName>
      <rasd:InstanceID>0</rasd:InstanceID>
      <rasd:ResourceType>4</rasd:ResourceType>
      <rasd:VirtualQuantity>512</rasd:VirtualQuantity>
  </Item>
  <Item ovf:bound="min">
      <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
      <rasd:ElementName>384 MB minimum memory size</rasd:ElementName>
      <rasd:InstanceID>0</rasd:InstanceID>
      <rasd:Reservation>384</rasd:Reservation>
```

```
            <rasd:ResourceType>4</rasd:ResourceType>
        </Item>
        <Item ovf:bound="max">
```

9 Основные разделы метаданных

В таблице 5 показаны основные разделы метаданных, которые определены.

Таблица 5 — Основные разделы метаданных

Раздел	Расположение	Число
DiskSection Описывает метаинформацию обо всех виртуальных дисках в пакете	Envelope	Ноль или одна
NetworkSection Описывает логические сети, используемые в пакете	Envelope	То же
ResourceAllocationSection Определяет резервирование, пределы и совместное использование ресурсов, таких как память или центральный процессор для набора виртуальных машин	VirtualSystemCollection	»
AnnotationSection Представляет собой аннотацию объекта в свободной форме	VirtualSystem VirtualSystemCollection	»
ProductSection Определяет информацию о продукте для пакета, такую как название и версия продукта, а также о наборе свойств, которые могут быть сконфигурированы	VirtualSystem VirtualSystemCollection	Ноль или более
EulaSection Определяет лицензионное соглашение для программного обеспечения в пакете	VirtualSystem VirtualSystemCollection	»
StartupSection Определяет, как запущен набор виртуальной машины	VirtualSystemCollection	Ноль или одна
DeploymentOptionSection Определяет дискретный набор необходимых потребностей в ресурсах	Envelope	То же
OperatingSystemSection Определяет установленную гостевую операционную систему виртуальной машины	VirtualSystem	»
InstallSection Определяет, должна ли производиться процедура первоначальной загрузки виртуальной машины для установки и конфигурации программного обеспечения	VirtualSystem	»

Следующие подразделы данного документа описывают семантику основных разделов и предлагают некоторые примеры. Разделы могут использоваться в нескольких местах конверта OVF. В описании каждого раздела указано, где он может использоваться. Подробную спецификацию всех атрибутов и элементов можно найти в схеме OVF.

В схеме OVF все разделы — часть группы подстановки с элементом Section в качестве заголовка группы замены. Элемент Section абстрактен и не может быть использован непосредственно.

9.1 Раздел дисков DiskSection

В разделе DiskSection размещается метаинформация о виртуальных дисках в пакете OVF. Виртуальные диски и их метаданные описаны вне виртуальных аппаратных средств, чтобы обеспечить совместное использование виртуальными машинами в рамках пакета OVF.

ПРИМЕР — Следующий пример демонстрирует описание виртуальных дисков.

```
<DiskSection>
 <Info>Описан набор виртуальных дисков </Info>
 <Disk ovf:diskId="vmdisk1" ovf:fileRef="file1" ovf:capacity="8589934592"
       ovf:populatedSize="3549324972"
       ovf:format=
         "gost.e-ecolog.ru                                                    ">
 </Disk>
 <Disk ovf:diskId="vmdisk2" ovf:capacity="536870912"
 </Disk>
 <Disk ovf:diskId="vmdisk3" ovf:capacity="${disk.size}"
       ovf:capacityAllocationUnits="byte * 2^30"
 </Disk>
</DiskSection>
```

DiskSection — это раздел, размещение которого допускается только на внешнем уровне конверта.

Каждый виртуальный диск представлен элементом Disk, которому нужно дать идентификатор, используя атрибут ovf:disked, причем идентификатор должен быть уникальным в пределах DiskSection.

Емкость виртуального диска должна быть определена атрибутом ovf:capacity с целочисленным значением xs:long. Единица выделения по умолчанию должна быть байтом. Для определения другой единицы выделения может быть использован дополнительный строковый атрибут ovf:capacityAllocationUnits. Значения ovf:capacityAllocationUnits должны соответствовать формату программных единиц, определенных в DSP0004 при том ограничении, что основной единицей должен быть «байт».

Атрибут ovf:fileRef указывает на содержимое виртуального диска, идентифицируя существующий элемент File в элементе ссылок References. Элемент File идентифицируется путем сопоставления значения его атрибута ovf:id со значением атрибута ovf:fileRef. Отсутствие атрибута ovf:fileRef должно обозначать пустой диск. В этом случае диск должен создаваться, и все содержимое диска обнулено во время установки. Обычно гостевое программное обеспечение будет форматировать пустые диски в определенном формате файловой системы.

Универсальный локатор ресурса URI формата (см. 5.2) непустого виртуального диска должен быть определен атрибутом ovf:format.

Разные элементы Disk не должны содержать атрибуты ovf:fileRef с идентичными значениями. Элементы Disk должны быть упорядочены таким образом, чтобы они идентифицировали любые элементы File в том же самом порядке, как они определены в элементе ссылок References.

Для пустых дисков вместо того чтобы определять фиксированный объем виртуального диска, емкость пустого диска может быть задана, используя свойство OVF, например, ovf:capacity =" ${disk.size}". Свойство OVF должно разрешить целочисленное значение xs:long. Описание свойств OVF приведено в 9.5. При использовании свойств OVF атрибут ovf:capacityAllocationUnits полезен по той причине, что пользователь может быть запрошен, и по запросу ввести информацию о емкости диска, например, в гигабайтах.

Для непустых дисков фактически используемый объем диска может быть определен дополнительно, с использованием атрибута ovf:populatedSize. Единица измерения этого атрибута всегда — байты. Атрибут ovf:populatedSize позволяет оценить объем используемого дискового пространства, но его значение не должно превышать ovf:capacity.

В разделе VirtualHardwareSection виртуальные дисковые устройства могут иметь элемент rasd:HostResource, ссылающийся на элемент Disk в DiskSection (см. 8.3). Объем виртуального диска должен быть определен атрибутом ovf:capacity элемента Disk. Если наряду с rasd:HostResource определен и элемент rasd:VirtualQuantity, то значение виртуального объема не должно приниматься во внимание и может иметь произвольное значение.

OVF позволяет представлять образ диска в виде набора блоков, измененных по сравнению с родительским образом. Использование родительских дисков, зачастую может существенно уменьшить размер пакета OVF в случае множества дисков с подобным содержанием. Для элемента Disk родительский диск может быть дополнительно определен путем использования атрибута ovf:parentRef, в котором должна содержаться действительная ссылка ovf:diskId на другой элемент

Disk. Если дисковый блок не существует локально, то производится поиск этого дискового блока в родительском диске. В разделе DiskSection родительские элементы Disk должны располагаться перед дочерними элементами Disk, которые ссылаются на них.

9.2 Раздел сетей NetworkSection

В разделе NetworkSection должны быть перечислены все логические сети, используемые в пакете OVF.

```
<NetworkSection>
 <Info> Список логических сетей, используемых пакетом </Info>
 <Network ovf:name="red">
  <Description>Сеть Red доступна для использования</Description>
 </Network>
 <Network ovf:name="blue">
  <Description>Сеть Blue доступна для использования</Description>
 </Network>
</NetworkSection>
```

NetworkSection является действительным элементом внешнего уровня конверта.

Все сети, на которые имеются ссылки в элементах Connection во всех элементах VirtualHardwareSection, должны быть определены в NetworkSection.

9.3 Раздел выделения ресурсов ResourceAllocationSection

В разделе ResourceAllocationSection описываются все требования выделения ресурсов для объекта VirtualSystemCollection. Эти требования ресурсов должны выполняться при развертывании пакета OVF.

```
<ResourceAllocationSection>
 <Info>Определяет резервирование ЦП и памяти для набора виртуальных машин </Info>
 <Item>
   <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
   <rasd:ElementName>300 MB reservation</rasd:ElementName>
   <rasd:InstanceID>0</rasd:InstanceID>
   <rasd:Reservation>300</rasd:Reservation>
    <rasd:ResourceType>4</rasd:ResourceType>
 </Item>
 <Item ovf:configuration="..." ovf:bound="...">
   <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
   <rasd:ElementName>500 MHz reservation</rasd:ElementName>
   <rasd:InstanceID>0</rasd:InstanceID>
   <rasd:Reservation>500</rasd:Reservation>
   <rasd:ResourceType>3</rasd:ResourceType>
 </Item>
</ResourceAllocationSection>
```

ResourceAllocationSection является действительным элементом объекта VirtualSystemCollection.

Дополнительный атрибут ovf:configuration содержит список имен конфигурации. Семантика этого атрибута для вариантов развертывания рассматривается в 9.8.

Дополнительный атрибут ovf:bound содержит значение min, max или normal.Семантика этого атрибута. описана в 8.4.

9.4 Раздел аннотации AnnotationSection

Элемент AnnotationSection — это определяемая пользователем аннотация объекта. Такие аннотации могут быть выведены на экран во время развертывания пакета OVF.

```
<AnnotationSection>
<Info> Аннотация сервиса. Может быть проигнорирована </Info>
<Annotation> В случае проблем свяжитесь с центром поддержки пользователей </Annotation>
</AnnotationSection >
```

Раздел AnnotationSection — это действительный элемент объектов VirtualSystem и VirtualSystemCollection.

В разделе 10 описаны детали локализации элемента аннотации Annotation.

9.5 Раздел продукта ProductSection

В разделе ProductSection дается информация о продукте для конкретной реализации, такая как название продукта, версия и поставщик.

```
<ProductSection ovf:class="com.mycrm.myservice" ovf:instance="1">
  <Info>Приводится информация о продукте для данного сервиса </Info>
  <Product>MyCRM Enterprise</Product>
  <Vendor>MyCRM Corporation</Vendor>
  <Version>4.5</Version>
  <FullVersion>4.5-b4523</FullVersion>
  <ProductUrl>gost.e-ecolog.ru                                    >
  <VendorUrl>gost.e-ecolog.ru                          >
  <Icon ovf:height="32" ovf:width="32" ovf:mimeType="image/png" ovf:fileRef="icon">
  <Category>Email properties</Category>
  <Property ovf:key="admin.email" ovf:type="string" ovf:userConfigurable="true">
    <Label>Admin email</Label>
    <Description>Email адрес администратора</Description>
  </Property>
  <Category>Admin properties</Category>
  <Property ovf:key="app.log" ovf:type="string" ovf:value="low"
ovf:userConfigurable="true">
    <Description>Уровень протоколирования для сервиса </Description>
  </Property>
  <Property ovf:key="app.isSecondary" ovf:value="false" ovf:type="boolean">
    <Description>Установка кластера для сервера приложений </Description>
  </Property>
  <Property ovf:key="app.ip" ovf:type="string" ovf:value="${appserver-vm}">
    <Description>IP-адрес сервера приложений виртуальных машин </Description>
  </Property>
</ProductSection>
```

Дополнительный элемент Product определяет название продукта, а дополнительный элемент Vendor определяет имя поставщика продукта. Дополнительный элемент Version определяет версию продукта в краткой форме, в то время как подробная форма версии продукта определяется значением дополнительного элемента FullVersion. Дополнительный элемент ProductUrl определяет ссылку URL, которая должна указывать на удобочитаемое описание продукта, а дополнительный элемент VendorUrl определяет ссылку URL, которая должна указывать на удобочитаемое описание поставщика.

Дополнительный элемент AppUrl определяет ссылку URL, указывающую на экземпляр развернутого продукта. Этот элемент введен в порядке эксперимента. Дополнительный элемент Icon определяет иконки продукта для дисплея. Этот элемент также введен в порядке эксперимента

Элементы свойства Property определяют параметры настройки уровня приложения и особенно важны для объектов, которые должны быть настроены во время развертывания с использованием конкретных параметров, таких как сетевые идентификационные данные, IP-адреса серверов DNS и шлюзов и других.

ProductSection — это действительный раздел объектов VirtualSystem и VirtualSystemCollection.

Элементы свойства Property могут быть сгруппированы путем использования элементов категории Category. Набор элементов Property, сгруппированных элементов Category, представляет собой последовательность элементов Property, следующих после элемента Category, исключая, однако, элементы, которые не являются элементами Property. Для пакетов OVF, содержащих большое число элементов Property, подобный подход может обеспечить более простую установку. Точно так же для каждого элемента Property может быть определена короткая метка путем определения его дочернего элемента метки Label в дополнение к описанию, определенному его дочерним элементом описания Description. Подробности локализации элемента категории Category и дочерних элементов Description и Label элемента Property рассмотрены в разделе10.

Каждому элементу Property в ProductSection необходимо, используя атрибут ovf:key, дать идентификатор, который уникален в пределах ProductSection.

Каждому элементу Property в ProductSection нужно задать тип, используя атрибут ovf:type, и дополнительно квалификаторы типа, используя атрибут ovf:qualifiers. Действительные типы перечислены в Таблице 6, а действительные квалификаторы — в Таблице 7.

Таблица 6 - Типы свойств (Property)

Тип	Описание
uint8	8-битовый целый без знака
sint8	8-битовый целый со знаком
uint16	16-битовый целый без знака
sint16	16-битовый целый со знаком
uint32	32-битовый целый без знака
sint32	32-битовый целый со знаком
uint64	64-битовый целый без знака
sint64	64-битовый целый со знаком
string	строковый
boolean	логический
real32	4-байтовый с плавающей точкой IEEE
real64	8-байтовый с плавающей точкой IEEE

Таблица 7 - Квалификаторы типов свойств

Тип	Описание
string	MinLen(min) Минимальная длина MaxLen(max) Максимальная длина ValueMap{...} Область значений
uint8 sint8 uint16 sint16 uint32 sint32 uint64 sint64	ValueMap{...} Область значений

Дополнительный атрибут ovf:value используется, чтобы обеспечить значение по умолчанию для свойства. Для определения альтернативных значений по умолчанию для конкретных конфигураций могут быть использованы один или более дополнительных элементов значения Value , как это определено в 9.8.

Дополнительный атрибут ovf:userConfigurable определяет, конфигурируемо ли значение свойства во время фазы установки. Если значение ovf:userConfigurable — ЛОЖЬ (FALSE) или опущено, то атрибут ovf:value определяет значение, которое будет использоваться во время установки для данного параметра настройки. Если же ovf:userConfigurable ИСТИНА (TRUE), то атрибут ovf:value определяет значение по умолчанию для того параметра настройки, который во время установки может быть изменен.

Простая реализация OVF, такая как установщик командной строки, обычно использует значения свойств по умолчанию и не выдает запрос даже в том случае, если значение ovf:userConfigurable ИСТИНА (TRUE). Для принудительной выдачи запроса во время запуска достаточно опустить атрибут ovf:value для целочисленных типов, потому что пустая строка не является действительным целочисленным значением. Для строковых типов запрос может быть инициирован добавлением квалификатора, требующего непустой строки (см. таблицу 7).

Дополнительный атрибут логического типа ovf:password указывает на то, что значение свойства может содержать конфиденциальную информацию. Значение атрибута по умолчанию — ЛОЖЬ (FALSE). При реализации OVF рекомендуется затенять значения свойств при запросе ввода в случае, если значение ovf:password задано как ИСТИНА (TRUE). Это подобно HTML-вводу текстового пароля. Заметьте, что данный механизм предоставляет собой всего лишь ограниченное средство обеспечения безопасности. Хотя конфиденциальные значения скрыты от случайного взгляда, однако, значения по умолчанию в дескрипторе OVF и присваиваемые значения в среде OVF все же передаются открытым текстом.

В пределах VirtualSystem или VirtualSystemCollection могут быть определены ноль или более разделов ProductSection. Как правило, ProductSection соответствует определенному установленному программному продукту. Каждый раздел продукта на одном определенном самом уровне объекта должен иметь уникальную пару атрибутов ovf:class и атрибута ovf:instance. В общем случае при наличии единственного раздела ProductSection используется, атрибуты ovf:class и ovf:instance являются не обязательными и по умолчанию имеют значения пустой строки. Рекомендуется использовать свойство ovf:class для однозначной идентификации программного продукта, применяя обратное соглашение о доменных именах. Примеры таких значений — com.vmware.tools и org.apache.tomcat. Если экземпляры одного продукта установлены многократно, то атрибут ovf:instance используется для идентификации отдельного экземпляра.

Элементы свойств Property передаются гостевому программному обеспечению через среду OVF, как описано в разделе 11. Значение атрибута ovfenv:key элемента Property, представленного в среде OVF, должно быть создано из значения атрибута ovf:key соответствующего элемента Property, определенного в объекте ProductSection дескриптора OVF, следующим образом:

key-value-env = [class-value "."] key-value-prod ["." instance-value]

где class-value — значение атрибута ovf:class элемента Property, определенного в объекте ProductSection. Сочетание [class-value"."] должно присутствовать тогда и только тогда, когда class-value не является пустой строкой;

-key-value-prod — значение атрибута ovf:key элемента Property, определенного в объекте ProductSection.

-instance-value — значение атрибута ovf:instance элемента Property, определенного в объекте ProductSection. Сочетание ["." instance-value] должно присутствовать тогда и только тогда, когда instance-value не является пустой строкой.

ПРИМЕР — В этом примере среды OVF показано, каким образом свойства могут быть переданы в гостевое программное обеспечение:

<Property ovf:key="com.vmware.tools.logLevel" ovf:value="none"/>

<Property ovf:key="org.apache.tomcat.logLevel.1" ovf:value="debug"/>

<Property ovf:key="org.apache.tomcat.logLevel.2" ovf:value="normal"/>

Потребитель пакета OVF должен запросить ввод свойств в случае, если ovf:userConfigurable — ИСТИНА (TRUE). Эти свойства могут быть определены в нескольких ProductSection, а также в подобъектах в пакета OVF.

Если ProductSection имеется, то в первом объекте ProductSection определенного элемента контента Content пакета верхнего уровня должна присутствовать информационная сводка, которая описывает весь пакет целиком. После установки по желанию потребителя пакета OVF эта информация может быть сделана доступной как экземпляр класса CIM_Product.

Элементы свойств Property, определенные в разделе VirtualSystemCollection, доступны также и непосредственными дочерними элементами этого раздела (см. раздел 11). Дочерние элементы могут обратиться к свойствам родительского VirtualSystemCollection путем использования макросов в форме ${имя} в качестве значений для атрибутов ovf:value.

Таблица 6 перечисляет действительные типы для свойств. Они составляют подмножество внутренних типов CIM, определенных в DSP0004, которое также определяет пространства значений и форматов для каждого внутреннего типа. Используя атрибут ovf:type, для каждого элемента Property должен быть определен тип.

В таблице 7 перечислены поддерживаемые квалификаторы типа CIM, как определено в DSP0004. Каждый элемент Property может дополнительно определить квалификаторы типа, используя атрибут ovf:qualifiers с несколькими квалификаторами, разделенными запятой (см. Создание списка квалификаторов qualifierList в ПРИЛОЖЕНИИ А «Описание синтаксиса грамматики MOF» в DSP0004.

9.6 Раздел лицензий EulaSection

Раздел лицензий EulaSection содержит юридические условия использования его родительского элемента контента Content. Эта лицензия должна быть показана и принята во время развертывания пакета OVF. В пакете OVF может быть несколько разделов EulaSection. Если допускаются установки без вмешательства персонала, то при установке неявно принимаются условия всех входящих в пакет разделов лицензий.

```
<EulaSection>
  <Info>Лицензионное соглашение</Info>
  <License>
Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor placerat
fermentum, enim integer ad vestibulum volutpat. Nisl rhoncus turpis est, vel elit,
congue wisi enim nunc ultricies sit, magna tincidunt. Maecenas aliquam maecenas ligula
nostra, accumsan taciti. Sociis mauris in integer, a dolor netus non dui aliquet,
sagittis felis sodales, dolor sociis mauris, vel eu libero cras. Interdum at. Eget
habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed
auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec
pellentesque leo, scelerisque.
  </License>
</EulaSection>
```

Раздел EulaSection является действительным разделом для объектов VirtualSystem и VirtualSystemCollection.

Подробности локализации элемента License приведены в разделе 10.

9.7 Раздел запуска StartupSection

Раздел StartupSection определяет, как должен стартовать и останавливаться набор виртуальных машин.

```
<StartupSection>
        <Item ovf:id="vm1" ovf:order="0" ovf:startDelay="30" ovf:stopDelay="0"
          ovf:startAction="powerOn" ovf:waitingForGuest="true"
ovf:stopAction="powerOff"/>
        <Item ovf:id="teamA" ovf:order="0"/>
        <Item ovf:id="vm2" ovf:order="1" ovf:startDelay="0" ovf:stopDelay="20"
          ovf:startAction="powerOn" ovf:stopAction="guestShutdown"/>
</StartupSection>
```

У каждого элемента контента Content, который является прямым дочерним элементом VirtualSystemCollection, может быть соответствующий элемент Item в объекте StartupSection объекта VirtualSystemCollection. Необходимо отметить, что элементы Item могут соответствовать и объектам VirtualSystem, и объектам VirtualSystemCollection. При выполнении запуска или остановки на объекте VirtualSystemCollection в указанном порядке вызывается выполнение соответствующих действий элемента Item его объекта StartupSection. Если элемент Item соответствует вложенному объекту VirtualSystemCollection, действия его элементов Item объекта StartupSection должны быть вызваны перед вызовом действий элемента Item, соответствующего объекту VirtualSystemCollection, т. е, осуществляется обход в глубину.

Для VirtualSystem и VirtualSystemCollection: поддерживаются следующие необходимые атрибуты элемента Item:

- атрибут ovf:id должен соответствовать значению атрибута ovf:id элемента Content, который является прямым дочерним элементом данного VirtualSystemCollection. Этот элемент Content описывает виртуальную машину или набор виртуальных машин, к которым применяются действия, определенные в элементе Item;

- атрибут ovf:order определяет порядок запуска, используя неотрицательные целочисленные значения. Порядок выполнения действий запуска это возрастающая последовательность чисел. Элементы Item с одним и тем же идентификатором порядка могут быть запущены одновременно. Порядок выполнения действия остановки—это убывающая числовая последовательность.

Для VirtualSystem поддерживаются следующие дополнительные атрибуты Элемента Item:

- атрибут ovf:startDelay определяет задержку в секундах для ожидания перехода к следующему по порядку действию в последовательности запуска. Значение по умолчанию 0;

- атрибут ovf:waitingForGuest позволяет платформе возобновить последовательность запуска после того как гостевое программное обеспечение сообщило, что это готово. Интерпретация этого атрибута определяется спецификой платформы развертывания. Значение по умолчанию — ЛОЖЬ (FALSE);

-атрибут ovf:startAction определяет используемое действие запуска. Действительные значения — включить питание(powerOn) и нет (none). Значение по умолчанию — включение питания (powerOn).

- атрибут ovf:stopDelay определяет задержку в секундах для ожидания завершения предыдущего по порядку действия в последовательности остановки. Значение по умолчанию 0;

- атрибут ovf:stopAction определяет используемое действие остановки. Действительные значения — выключение питания (powerOff), останов гостевого программного обеспечения (guestShutdown) и нет (none). Интерпретация guestShutdown определяется спецификой платформы развертывания. Значение по умолчанию — выключение питания(powerOff).

Если ничего не задано, то для каждого объекта в наборе создается неявный элемент Item по умолчанию в сочетании ovf:order = "0". Таким образом, для тривиальной последовательности запуска не требуется определения StartupSection.

9.8 Раздел опций развертывания DeploymentOptionSection

Раздел опций развертывания DeploymentOptionSection определяет дискретный набор предполагаемых конфигураций ресурсов. Автор пакета OVF может включать метаданные настройки для различных конфигураций. Потребитель OVF должен выбрать конфигурацию, например, запрашивая пользователя. Выбранная конфигурация видима в среде OVF(см. раздел 11), позволяя таким образом адаптацию гостевого программного обеспечения к выбранной конфигурации. Раздел DeploymentOptionSection определяет идентификатор, метку и описание для каждой конкретной конфигурации.

```
<DeploymentOptionSection>
    <Configuration ovf:id="Minimal">
        <Label>Minimal</Label>
        <Description>Некоторое описание</Description>
    </Configuration>
    <Configuration ovf:id="Typical" ovf:default="true">
        <Label>Typical</Label>
        <Description> Некоторое описание </Description>
    </Configuration>
    <!—Дополнительные конфигурации -->
</DeploymentOptionSection>
```

В разделе DeploymentOptionSection используется следующая семантика.

- Раздел DeploymentOptionSection в случае, когда он есть, действителен только на уровне конверта, и только один раздел должен быть определен в дескрипторе OVF.

- Дискретный набор конфигураций описывается посредством элементов конфигурации Configuration, для которых должны определить уникальные в пакете атрибуты идентификаторов ovf:id.

- Элемент конфигурации Configuration по умолчанию может быть определен с помощью дополнительного атрибута ovf:default. Если значение по умолчанию не определено, то первый элемент в списке считается значением по умолчанию. Определение более одого элемента в качестве значения по умолчанию недопустимо.

- Элементы меток Label и описаний Description локализуемы с использованием атрибута ovf:msgid. Более детально поддержка интернационализации рассмотрена в разделе 10.

Конфигурации могут использоваться для управления ресурсами как для виртуальных аппаратных средств, так и для наборов виртуальных машин. Элементы Item в элементах

VirtualHardwareSection описывают ресурсы объектов для виртуальных систем VirtualSystem, в то время как элементы Item в элементах ResourceAllocationSection описывают ресурсы для наборов виртуальных машин. Для этих двух типов элемента Item определена следующая дополнительная семантика:

- Каждый элемент Item имеет дополнительный атрибут конфигурации ovf:configuration, содержащий список конфигураций, разделенных одним символом пробела. Если атрибут не определен, то элемент должен использоваться для любой конфигурации, а если определен, то элемент должен выбраться только в том случае, если в списке присутствует идентификатор выбранной конфигурации. Атрибут конфигурации не должен содержать идентификаторов, не определенных в DeploymentOptionSection.

- Допускается, чтобы в рамках одного VirtualHardwareSection или одного ResourceAllocationSection несколько элементов Item ссылались на один и тот же идентификатор объекта InstanceID. Единый объединенный Элемент Item для данного InstanceID должен создаваться посредством сбора дочерних элементов каждого элемента Item с дочерними элементами прежнего элемента Item в дескрипторе OVF, не принимая во внимание случаи подобно-именованных дочерних элементов в последнем элементе Item. Любые атрибуты, определенные для дочерних элементов элемента Item, которые проигнорированы подобным образом, не являются частью объединенного элемента Item.

- Все элементы Item должны специфицировать ResourceType, а элементы Item с одним и тем же InstanceID должны быть согласованы с ResourceType.

ПРИМЕР 1 — В следующем примере показан VirtualHardwareSection:

```
<VirtualHardwareSection>
        <Info>...</Info>
        <Item>
          <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
          <rasd:ElementName>512 MB memory size and 256 MB reservation
    </rasd:ElementName>
          <rasd:InstanceID>0</rasd:InstanceID>
          <rasd:Reservation>256</rasd:Reservation>
          <rasd:ResourceType>4</rasd:ResourceType>
          <rasd:VirtualQuantity>512</rasd:VirtualQuantity>
        </Item>
        ...
        <Item ovf:configuration="big">
          <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
          <rasd:ElementName>1024 MB memory size and 512 MB reservation
    </rasd:ElementName>
          <rasd:InstanceID>0</rasd:InstanceID>
          <rasd:Reservation>512</rasd:Reservation>
          <rasd:ResourceType>4</rasd:ResourceType>
          <rasd:VirtualQuantity>1024</rasd:VirtualQuantity>
        </Item>
</VirtualHardwareSection>
```

Необходимо отметить, что использование комбинации атрибутов ovf:configuration и ovf:bound элемента Item обеспечивает повышенную гибкость параметров конфигурации.

Конфигурации могут далее использоваться, чтобы управлять значениями свойств по умолчанию. Для элементов свойства Property в ProductSection определена следующая дополнительная семантика:

- Можно использовать альтернативные значения свойств по умолчанию для различных конфигураций в DeploymentOptionSection. Помимо элемента Label и Description, каждый элемент свойства может дополнительно содержать элементы значения Value. У элемента значения Value должен быть атрибут ovf:value, определяющий альтернативное значение по умолчанию, и атрибут

ovf:configuration, определяющий конфигурацию, в которой должно использоваться это новое значение по умолчанию. Несколько элементов Value не должны ссылаться на одну и ту же конфигурацию.

ПРИМЕР 2 — Далее показан раздел ProductSection:

```
<ProductSection>
  <Property ovf:key="app.log" ovf:type="string" ovf:value="low"
ovf:userConfigurable="true">
      <Label>Loglevel</Label>
      <Description>Loglevel for the service</Description>
      <Value ovf:value="none" ovf:configuration="minimal">
  </Property>
</ProductSection>
```

9.9 Раздел операционной системы OperatingSystemSection

Раздел OperatingSystemSection определяет операционную систему, установленную на виртуальной машине.

```
<OperatingSystemSection ovf:id="76">
  <Info>Определяет установленную операционную систему </Info>
  <Description>Microsoft Windows Server 2008</Description>
</OperatingSystemSection>
```

Действительные значения для ovf:id определены квалификатором ValueMap свойства CIM_OperatingSystem.OsType.

OperatingSystemSection является действительным разделом только для объекта VirtualSystem.

9.10 Раздел установки InstallSection

Если раздел InstallSection не определен, то это означает, что для завершения установки гостевого программного обеспечения нет необходимости включения виртуальной машины.

```
<InstallSection ovf:initialBootStopDelay="300">
```

<Info> Определяет, что для установки и конфигурации программного обеспечения виртуальная машина должна быть перезагружена после создания гостевого программного обеспечения

```
  </Info>
</InstallSection>
```

Раздел InstallSection, если таковой определен, указывает на то, что для установки и/или конфигурирования гостевого программного обеспечения виртуальная машина должна быть один раз перезагружена. Гостевое программное обеспечение, как ожидают, получит доступ к среде OVF во время этой начальной загрузки и завершит работу после окончания установки и/или конфигурации программного обеспечения, а также завершит работу гостя.

10 Интернационализация

Использованием дополнительного атрибута ovf:msgid поддерживается локализация сообщений следующих элементов:

- элемент информации Info в контенте Content;
- элемент имени Name в контенте Content;
- элемент информации Info в разделе Section;
- элемент аннотации Annotation в разделе AnnotationSection;
- элемент лицензии License в разделе EulaSection;
- элемент описания Description в разделе NetworkSection;
- элемент описания Description в разделе OperatingSystemSection;
- элементы Description, Product, Vendor, Label и Category в разделе ProductSection;
- элементы описания Description и метки Label в свойстве Property;
- элементы описания Description и метки Label в разделе DeploymentOptionSection;

- подэлементы ElementName, Caption и Description элемента системы System в разделе VirtualHardwareSection;
- подэлементы ElementName, Caption и Description элемента Item в разделе VirtualHardwareSection;
- подэлементы ElementName, Caption и Description элемента Item в разделе inResourceAllocationSection.

Атрибут ovf:msgid содержит идентификатор, который ссылается на сообщение, которое может иметь различные значения для различных локалий.

Пример 1

<Info ovf:msgid="info.text">Значение текста info.text по умолчанию, если локаль отсутствует или не соответствует </Info>

<License ovf:msgid="license.tomcat-6_0"/> <!—Сообщения по умолчанию - нет -->

Атрибут xml:lang элемента конверт Envelope должен определить локаль по умолчанию для сообщений в дескрипторе. Атрибут является дополнительным со значением по умолчанию «en-US».

Пакеты ресурсов обработки сообщений могут быть внутренними или внешними к дескриптору OVF. Внутренние пакеты ресурсов представлены как элементы Strings в конце элемент конверта Envelope.

Пример 2

```
<ovf:Envelope xml:lang="en-US">
  ...
  ... здесь размещаются разделы и контент ...
  ...
  <Info msgid="info.os">Operating System</Info>
  ...
  <Strings xml:lang="da-DA">
       <Msg ovf:msgid="info.os">Operativsystem</Msg>
       ...
  </Strings>
  <Strings xml:lang="de-DE">
       <Msg ovf:msgid="info.os">Betriebssystem</Msg>
       ...
  </Strings>
</ovf:Envelope>
```

Внешние пакеты ресурсов должны быть перечислены в начале раздела ссылок References, а элементы Strings должны на них ссылаться. Внешний пакет обработки сообщений следует той же самой схеме, как и встроенный. Во внешнем пакете обработки сообщений точно один элемент Strings должен присутствовать во внешнем пакете сообщения, и элемент Strings может не иметь определенного атрибута ovf:fileRef.

Пример 3

```
<ovf:Envelope xml:lang="en-US">
  <References>
       ...
       <File ovf:id="it-it-resources" ovf:href="resources/it-it-bundle.msg"/>
  </References>
   ... здесь размещаются разделы и контент ...
   ...
   <Strings xml:lang="it-IT" ovf:fileRef="it-it-resources"/>
       ...
</ovf:Envelope>
```

Пример 4 – Пример содержимого внешнего файла resources/it-it-bundle.msg на который ссылается предыдущий пример:

```
<Strings
  xmlns:ovf="gost.e-ecolog.ru                                    "
  xmlns="gost.e-ecolog.ru                                    "
  xml:lang="it-IT">
      <Msg ovf:msgid="info.os">Sistema operativo</Msg>
      ...
</Strings>
```

11 Среда OVF (OVF Environment)

Среда OVF определяет, как взаимодействуют гостевое программное обеспечение и платформа развертывания. Эта среда позволяет гостевому программному обеспечению получать доступ к информации о платформе развертывания, такой как определенные пользователем значения для свойств, определенных в дескрипторе OVF. Спецификация среды поделена на раздел *протокола protocol* и раздел *транспорта transport*. Раздел *протокола* определяет формат и семантику XML-документа, который может быть сделан доступным для гостевого программного обеспечения. Раздел т*ранспорта* определяет, как передается информация между платформой развертывания и гостевым программным обеспечением.

В файле определения XML-схемы dsp8027_1.1.0.xsd для среды OVF перечислены элементы и атрибуты.

11.1 Документ среды Environment Document

Документ среды – это расширяемый XML-документ, который обеспечивает гостевое программное обеспечение информацией среды выполнения. Способ, посредством которого этот документ доставляется, зависит от типа транспорта.

ПРИМЕР – В следующем примере приводится структура документа среды.

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment xmlns:xsi="gost.e-ecolog.ru                                    "
      xmlns:ovfenv="gost.e-ecolog.ru                                  "
      xmlns="gost.e-ecolog.ru                                  "
      ovfenv:id="идентификатор виртуальной машины из дескриптора OVF">
  <!—Информация о платформе виртуализации –>
  <PlatformSection>
    <Kind> Тип платформы виртуализации </Kind>
    <Version>Версия платформы виртуализации</Version>
    <Vendor> Поставщик платформы виртуализации</Vendor>
    <Locale>Язык и код страны</Locale>
    <TimeZone>Текущая разница во времени с UTC в минутах</TimeZone>
  </PlatformSection>
  <!--- Свойства, определенные для данной виртуальной машины -->
    <PropertySection>
     <Property ovfenv:key="key" ovfenv:value="value">
     <!—еще свойства –>
    </PropertySection>
    <Entity ovfenv:id="Идентификатор одноуровневой виртуальной системы или
набора виртуальных систем ">
      <PropertySection>
        <!—Свойства с родственных систем -->
      </PropertySection>
      </Entity>
</Environment>
```

Значение ovfenv:id атрибута элемента среды Environment должно соответствовать значению ovf:id атрибута объекта VirtualSystem, описывающего эту виртуальную машину.

Элемент PlatformSection содержит дополнительную информацию, предоставленную платформой развертывания. Элементы Kind, Version, и Vendor описывают детали поставщика платформы развертывания. Эти элементы введены в порядке эксперимента. Элементы Locale и TimeZone описывают текущую локаль и часовой пояс; они также введены в порядке эксперимента.

Элемент PropertySection содержит элементы Property с парами ключ/значение, соответствующие всем свойствам, определенным в дескрипторе OVF для текущей виртуальной машины, а также всем свойствам, определенным для непосредственного родителя VirtualSystemCollection, если такой имеется. Для облегчения анализа приложениями среда представляет свойства в виде простого списка. Кроме того, формат единого списка поддерживает семантику переопределения, когда свойство из VirtualSystem может заменить свойство, определенное родительским VirtualSystemCollection. Свойство, которое было переопределено, не должно находиться в списке. Переопределение может произойти, если у свойства в текущей виртуальной машине и у свойства в родительском VirtualSystemCollection идентичны значения атрибутов ovf:key, ovf:class и ovf:instance (см. 9.5). В этом случае значение переопределенного родительского свойства может быть получено добавлением дочернего свойства с другим именем, ссылающегося на родительское свойство посредством макроса (см. 9.5).

Элемент объекта Entity должен существовать для каждого элемента VirtualSystem и VirtualSystemCollection одного уровня, если такие имеются. Значение ovfenv:id атрибута элемента объекта Entity должно соответствовать значению ovf:id атрибута объекта того же уровня. Элементы объекта Entity содержат пары свойств key/value в документах среды OVF одного уровня таким образом, что контент элемента объекта Entity для определенного уровня должен содержать точный PropertySection, доступный на этом уровне. Эта информация может использоваться, чтобы сделать информацию о конфигурации, например, такую как IP-адреса, доступной для систем VirtualSystems, являющейся частью многоуровневого приложения.

В таблице 8 показаны основные разделы.

Таблица 8 – Основные разделы

Раздел	Расположение	Число
PlatformSection Обеспечивает информацией от платформы развертывания	Среда (Environment)	Ноль или один
PropertySection Содержит пары key/value, соответствующие свойствам, определенным в дескрипторе OVF	Среда (Environment) Объект (Entity)	То же

Документ среды расширяем и позволяет добавлять разделы новых типов. Потребителю документа следует игнорировать разделы и элементы неизвестного типа.

11.2 Раздел транспорта (Transport)

Информация документа среды может быть передана гостевому программному обеспечению многими способами. Эти способы называют типами транспорта. Типы транспорта определяются в дескрипторе OVF атрибутом ovf:transport раздела VirtualHardwareSection. Можно определить несколько типов транспорта, разделенных одним символом пробела. В этом случае реализация может использовать любой из них. Типы транспорта определяют методы, которыми документ среды передается гостевому программному обеспечению платформой развертывания.

Для обеспечения функциональной совместимости данная спецификация определяет тип транспорта «iso», который должны поддерживать все реализации, обеспечивающие поддержку устройства CD-ROM. Транспорт iso, взаимодействуя с документом среды, делает динамически сгенерированный образ диска ISO доступным для гостевого программного обеспечения. С целью поддержки типа транспорта iso реализация должна до начальной загрузки виртуальной машины сделать образ диска ISO доступным только для чтения в качестве резервного для отсоединенного CD-ROM. Если для VirtualHardwareSection выбирается транспорт iso, то в этом разделе должно присутствовать, по крайней мере, одно неприсоединенное устройство CD-ROM.

Сгенерированный образ ISO должен соответствовать спецификации ISO 9660 с поддержкой расширений Joliet.

Изображение ISO должно содержать среду OVF для данной конкретной виртуальной машины, и эта среда должна размещаться в XML-файле с названием ovf-env.xml, который содержится в

корневой директории образа ISO. Таким образом, гостевое программное обеспечение может получить доступ к информации, используя стандартные инструменты гостевой операционной системы.

Если виртуальная машина до начальной загрузки имела более одного неприсоединенного CD-ROM, то гостевому программному обеспечению, вероятно, придется отсканировать присоединенные устройства CD-ROM для определения расположения образа ISO, содержащего ovf-env.xml — файл.

Образ диска ISO, содержащий среду OVF, должен быть доступен для гостевого программного обеспечения при каждой начальной загрузке виртуальной машины.

Поддержка типа транспорта «iso» не может быть требованием для архитектуры виртуального аппаратного обеспечения или гостевых операционных систем, у которых нет поддержки устройства CD-ROM.

Для совместимости с настоящей спецификацией при использовании любого отличного от iso формата транспорта обязательно должен быть представлен унифицированный идентификатор ресурса URI, который идентифицирует полноценную спецификацию того, как использовать транспорт. Спецификация не должна быть машиночитаемой, но должна быть статичной и уникальной таким образом, чтобы она могла использоваться программным обеспечением в качестве ключа при считывании дескриптора OVF для однозначного определения формата. Спецификация должна быть достаточной для интерпретации транспортного механизма квалифицированным человеком для реализации протоколов. Рекомендуется, чтобы эти URI были разрешимы.

Приложение А
(информативное)

Символы и соглашения

Примеры XML используют префиксы пространства имен XML, определенные в таблице 1. Примеры XML не определяют префиксы пространства имен для дочернего элемента. Необходимо отметить, что правила XML определяют, что, дочерние элементы, специфицированные без префикса пространства имен, используют префиксы пространства имен родительского элемента, а не пространства имен по умолчанию из XML-документа. Во всем документе пробел в значениях элементов XML использован для удобства чтения. На практике, сервисы могут воспринимать и удалять начальные и конечные пробелы в значениях элементов так, как будто этих пробелов нет.

Определения синтаксиса в расширенной нотации BNF (ABNF) используют ABNF как определено в IETF RFC5234 со следующими исключениями:

- Правила, разделенные вертикальной чертой (|), представляют возможность выбора, вместо использования наклонную черту вправо (/) как определено в ABNF.
- Любые символы должен рассматриваться, как регистрозависимые, в отличие от независимости от регистра как определено в ABNF.
- Пробел (то есть, символ пробела U+0020 и символ табуляции U+0009) допускается в синтаксических элементах, вместо того, чтобы элементы собирались бы без пробела как определено в ABNF.

Приложение В
(информативное)

Протокол изменений

Версия	Дата	Описание
1.0.0	2009-02-22	Стандарт DMTF
1.1.0	2010-01-12	Стандарт DMTF

Приложение С
(нормативное)

OVF XSD

Нормативные копии XML-схем для данной спецификации могут быть получены при использовании следующих ссылок URL:

gost.e-ecolog.ru

gost.e-ecolog.ru 1.1.0.xsd

Любой контент xs:documentation в XML-схемах для этой спецификации информативен и предназначен только для удобства чтения.

Нормативные копии XML-схем для отображения WS-CIM, (DSP0230) CIM_ResourceAllocationSystemSettingsData и CIM_VirtualSystemSettingData могут быть получены, используя следующие ссылки URL:

gost.e-ecolog.ru

gost.e-ecolog.ru _ResourceAllocationSettingData.xsd

Данная спецификация основана на следующих CIM MOF: CIM_VirtualSystemSettingData.mof
CIM_ResourceAllocationSettingData.mof
CIM_OperatingSystem.mof

Приложение Г
(информативное)

Библиография

ISO 9660, Спецификация расширений Joliet, Май 1995

W3C, Y. Savourel et и др., Лучшие практики интернационализации для XML,Рабочий проект, Октябрь *2007, http://www.w3.org/TR/2007/WD-xml-i18n-bp-20071031*

DMTF DSP1044,Профиль виртуализации процессорного устройства 1.0
http://www.dmtf.org/standards/published documents/DSP1044 1.0.pdf

DMTF DSP1045, Профиль виртуализации ресурсов памяти 1.0
gost.e-ecolog.ru

DMTF DSP1047, Профиль виртуализации устройств хранения данных 1.0
gost.e-ecolog.ru

DMTF DSP1022, Профиль ЦП 1.0,
gost.e-ecolog.ru

DMTF DSP1026, Профиль системной памяти 1.0,
gost.e-ecolog.ru

DMTF DSP1014, Профиль порта Ethernet 1.0,
gost.e-ecolog.ru

DSP1050, Профиль виртуализации ресурсов порта Ethernet 1.0
*http://*gost.e-ecolog.ru *documents/DSP1050 1.0.pdf*

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации

Таблица ДА 1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО 9660	–	*
Директивы ИСО/МЭК часть 2.1	–	*
ANSI/IEEE 1003.1	–	*
DMTF CIM 2.22	–	*
DMTF DSP0004	–	*
DMTF DSP0230	–	*
DMTF DSP1041	–	*
DMTF DSP1043	–	*
IETF RFC1738	–	*
IETF RFC1952	–	*
IETF RFC5234	–	*
IETF RFC2616	–	*
IETF RFC3986		
*Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта.		

УДК: 681.3.01:006.354 ОКС: 35.100.05

Ключевые слова: открытый формат визуализации, интероперабельность

Подписано в печать 01.04.2014. Формат 60x84 1/8.
Усл. печ. л. 4,65. Тираж 31 экз. Зак. 1607

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ФГУП «СТАНДАРТИНФОРМ»

123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru