文書番号: DSP0243
日付: 2010 年 1 月 12 日
バージョン: 1.1.0

オープン仮想化フォーマット仕様

文書タイプ: 仕様

文書の位置づけ: DMTF標準

文書の言語: 英語

目次

序文 5
はじめに 5
1 範囲 7
2 規範的参照 7
3 良いウゴと定義 8
4 シンボルと略語 10
5 OVF パッケージ 10
5.1 OVF パッケージ構造 10
5.2 仮想ディスクフォーマット 12
5.3 シングルファイルとしての配布 12
5.4 ファイルセットとしての配布 13
6 OVF 記述子 13
7 エンベロープエレメント 14
7.1 ファイルリファレンス 15
7.2 コンテンツエレメント 16
7.3 拡張性 17
7.4 適合性 18
8 仮想ハードウェア記述子 19
8.1 VirtualHardwareSection 19
8.2 拡張性 20
8.3 仮想ハードウェアエレメント 20
8.4 エレメント上の範囲 22
9 コアメタデータセクション 24
9.1 DiskSection 25
9.2 NetworkSection 26
9.3 ResourceAllocationSection 26
9.4 AnnotationSection 27
9.5 ProductSection 27
9.6 EulaSection 30
9.7 StartupSection 31
9.8 DeploymentOptionSection 32
9.9 OperatingSystemSection 34
9.10 InstallSection 34
10 国際化 34
11 OVF 環境 35

11.1 環境文書 36
11.2 トランスポート 38
附属書A（情報提供）シンボルと規約 39
附属書B（情報提供）変更履歴 40
附属書C（情報提供）OVF XSD 41
参考文献 42

表

表 1－XML ネームスペースプレフィクス 13
表 2－ovf:required 属性を持つ子エレメントの動作 20
表 3－HostResource エレメント 21
表 4－仮想デバイスおよびコントローラのエレメント 22
表 5－コアメタデータセクション 24
表 6－プロパティタイプ 30
表 7－プロパティ修飾子 30
表 8－コアセクション 38

序文

オープン仮想化フォーマット仕様（The *Open Virtualization Format Specification*）（DSP0243）は、DMTF のシステム仮想化、パーティショニング、クラスタリングワーキンググループによって作成された。

本仕様は、下記の多くの個人およびチームの共同作業の結果として発展してきた。

Simon Crosby, XenSource

Ron Doyle, IBM

Mike Gering, IBM

Michael Gionfriddo, Sun Microsystems

Steffen Grarup, VMware (Co-Editor)

Steve Hand, Symantec

Mark Hapner, Sun Microsystems

Daniel Hiltgen, VMware

Michael Johanssen, IBM

Lawrence J. Lamers, VMware (Chair)

John Leung, Intel Corporation

Fumio Machida, NEC Corporation

Andreas Maier, IBM

Ewan Mellor, XenSource

John Parchem, Microsoft

Shishir Pardikar, XenSource

Stephen J. Schmidt, IBM

René W. Schmidt, VMware (Co-Editor)

Andrew Warfield, XenSource

Mark D. Weitzel, IBM

John Wilson, Dell

はじめに

オープン仮想化フォーマット（OVF）仕様は、仮想マシンで実行されるソフトウェアのパッケージ化および配布を目的とした、オープンかつ安全で、可搬性、効率性、拡張性に優れたフォーマットを記述する。フォーマットの主要なプロパティは次の通りである。

・ 配布に最適化

OVF は業界標準のパブリックキーインフラストラクチャーに基づき、コンテンツの検証とインテグリティのチェックをサポートし、ソフトウェアライセンシングの基本的な管理戦略を提供する。

・ シンプルで自動的なユーザエクスペリエンスに最適化

OVF は仮想マシン（VM）のライフサイクル管理プロセスの導入段階における、OVF のパッケージ全体、各仮想マシンまたはメタデータコンポーネントの検証をサポートする。OVF はユーザが読み取れる形のパッケージ関連記述情報もパッケージ化し、この情報により仮想化プラットフォームはインストールエクスペリエンスを効率化することができる。

・ シングル VM、複数 VM コンフィギュレーションの両方をサポート

OVF は標準シングル VM パッケージと、相互依存する複数の VM から成る複合型多層サービスを持つパッケージの、両方をサポートする。

・ 可搬性のある VM パッケージ化

OVF は仮想化プラットフォームに中立的な対場をとるが、OVF によってプラットフォーム特有の機能強化も可能になる。OVF は現在ハイパーバイザに使用されている仮想化ハードディスクフォーマット全範囲をサポートしているが、OFV は拡張可能であるため、将来発生するフォーマットにも対応できる。仮想マシンのプロパティは簡潔、正確にキャプチャされる。

・ ベンダおよびプラットフォーム独立

OVF は特定のホストプラットフォーム、仮想化プラットフォームまたはゲストオペレーティングシステムに依存しない。

・ 拡張可能

OVF は直ちに役立ち、拡張可能である。OVF は、業界の仮想化アプライアンス技術の推進に伴い、拡張できるように設計されている。OVF は、特定業種をサポートするベンダ特有のメタデータのエンコーディングもサポートし、許可する。

・ ローカライズ可能

OVF は複数のロケールで、ユーザに見える形の記述をサポートし、アプライアンスのインストール中に、やりとりプロセスのローカライゼーションをサポートする。この機能によって、シングルパッケージ化されたアプライアンスが複数の市場機会に応えることができる。

・オープン標準
OVF は業界の主要ベンダの協力から生まれ、可搬性のある仮想マシンの将来の標準として、業界が承認したフォーラムで発展している。

効率的な実行フォーマットとなることが OVF の明確な目的ではない。オープン仮想化フォーマットから取り出した仮想マシンのソフトウェアを動作するために、ハイパーバイザが許可されているが、必要ではない。

オープン仮想化フォーマット仕様
1　範囲
オープン仮想化フォーマット（OVF）仕様は、仮想マシンで動作するソフトウェアのパッケージ化と配布を目的とした、オープンかつ安全で、可搬性、効率性、拡張性に優れたフォーマットを記述する。

2　規範的参照
次の参照文書は本文書の適用に不可欠である。日付のある文書に関しては、引用した判のみが適用される。日付のない文書に関しては、参照文書の最新版が（修正も含めて）適用される。

ANSI/IEEE Standard 1003.1-2008, *IEEE Standard for Information Technology- Portable Operating System Interface (POSIX) Base Specifications, Issue 7*, Institute of Electrical and Electronics Engineers, December 2008, http://standards.ieee.org/index.html

DMTF CIM Schema 2.22, http://www.dmtf.org/standards/cim

DMTF DSP0004, *Common Information Model (CIM) Infrastructure Specification 2.5*, http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

DMTF DSP0230, *WS-CIM Mapping Specification 1.0*, http://www.dmtf.org/standards/published_documents/DSP0230_1.0.pdf

DMTF DSP1041, *Resource Allocation Profile (RAP) 1.1*, http://www.dmtf.org/standards/published_documents/DSP1041_1.1.pdf

DMTF DSP1043, *Allocation Capabilities Profile (ACP) 1.0*, http://www.dmtf.org/standards/published_documents/DSP1043_1.0.pdf

IETF RFC 1738, T. Berners-Lee, *Uniform Resource Locators (URL)*, December 1994, http://www.ietf.org/rfc/rfc1738.txt

IETF RFC 1952, P. Deutsch, *GZIP file format specification version 4.3*, May 1996, http://www.ietf.org/rfc/rfc1952.txt

IETF RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*, http://www.ietf.org/rfc/rfc5234.txt

IETF RFC 2616, R. Fielding et al, *Hypertext Transfer Protocol – HTTP/1.1*, June 1999, http://www.ietf.org/rfc/rfc2616.txt

IETF RFC 3986, *Uniform Resource Identifiers (URI): Generic Syntax*, http://www.ietf.org/rfc/rfc3986.txt

ISO 9660, 1988 Information processing-Volume and file structure of CD-ROM for information interchange, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=17505

ISO, ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*, http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

3 用語と定義

本文書では、次の用語と定義を適用する。

3.1

できる（can）

物質的、物理的、因果的のいずれかの可能性および能力を表すときに使う。

3.2

できない（cannot）

物質的、物理的、因果的のいずれかの可能性および能力を表すときに使う。

3.3

条件付（conditional）

特定の条件を満たす場合に、文書に準拠するために厳密に従うべき要件を示す。

3.4

義務（mandatory）

文書に準拠するために厳密に従うべき要件を示し、そこからの逸脱はまったく認められない。

3.5

してもよい（may）

本文書の範囲内で許容される行動を示す。

3.6
する必要はない（need not）
本文書の範囲内で許容される行動を示す。
3.7
任意である（optional）
本文書の範囲内で許容される行動を示す。
3.8
するものとする（shall）
文書に準拠するために厳密に従うべき要件を示し、そこからの逸脱はまったく認められない。
3.9
しないものとする（shall not）
文書に準拠するために厳密に従うべき要件を示し、そこからの逸脱はまったく認められない。
3.10
すべきである（should）
いくつかの可能性の中から、他の可能性に言及することや排除すること無く、特にふさわしい 1 つの可能性が推奨されることを示し、また、ある行動が好ましいが必ずしも必要ではないことを示す。
3.11
すべきではない（should not）
ある行動の可能性または方針は推奨されないが、禁じられてはいないことを示す。
3.12
アプライアンス
仮想アプライアンスを参照。
3.13
デプロイメントプラットフォーム
OVF パッケージをインストールする製品。
3.14
ゲストソフトウェア
仮想マシンの電源をオンにすると動作する、仮想化ディスク上に保存されたソフトウェアである。ゲストは通常、オペレーティングシステム、ユーザレベルのアプリケーションとサービスである。
3.15
OVF パッケージ
ゼロ以上のファイルを伴う、OVF XML 記述子ファイル。

3.16
OVF 記述子
OVF XML 記述子ファイル。
3.17
プラットフォーム
デプロイメントプラットフォームを参照。
3.18
仮想アプライアンス
1 以上の仮想マシンにインストールされた完全なソフトウェアスタックとして提供されるサービス。仮想アプライアンスは通常、OVF パッケージ内で提供されると期待されている。
3.19
仮想ハードウェア
ゲストソフトウェアから見えるハードウェア（CPU、コントローラ、イータネットデバイス、ディスクを含む)。
3.20
仮想マシン
ゲストソフトウェアの実行をサポートする完全な環境。
仮想マシンは、仮想ハードウェア、仮想ディスクおよびそれらに関連するメタデータをすべて含む。仮想マシンを使うことによって、ハイパーバイザと呼ばれるソフトウェア層を通じて、根底となる物理マシンの多重化ができる。
3.21
仮想マシンコレクション
仮想マシンのセットから成るサービス。サービスは 1 以上の仮想マシンから成るシンプルなセットでもよく、または仮想マシンと別の仮想マシンコレクションとの組み合わせから構築される複合型サービスであってもよい。仮想マシンコレクションを構成することができるので、複合型のネストされたコンポーネントが可能になる。

4. シンボルと略語
本文書では次のシンボルと略語を使用している。
4.1.1
CIM
共通情報モデル
4.1.2
IP
インターネットプロトコル
4.1.3

OVF
オープン仮想化フォーマット
4.1.4
VM
仮想マシン

5 OVF パッケージ

5.1 OVF パッケージ構造

OVF パッケージは次のファイルで構成されるものとする。

- 1 つの OVF 識別子。拡張子は.ovf
- ゼロまたは 1 の OVF マニフェスト。拡張子は.mf
- ゼロまたは 1 の OVF 証明書。拡張子は.cert
- ゼロ以上のディスクイメージファイル。
- ゼロ以上の、ISO イメージ等の追加リソースファイル。

.ovf、.mf、.cert のファイル拡張子を使用するものとする。

例1：次のファイルリストはOVFパッケージの1例である。

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

注：前出の例はVMDKディスクファイルを使用しているが、複数ディスクフォーマットもサポートされている。

OVF パッケージは 5.3 および 5.4 に記載の通り、シングルユニットまたはファイルセットのいずれかとして保存することができる。どちらのモードもサポートされるものとする。

OVF パッケージが持つマニフェストファイルが、OVF パッケージに含まれる個々のファイルの SHA-1 ダイジェストを格納するマニフェストファイルであることもある。マニフェストファイルは、拡張子.mf と、.ovf ファイルと同じベースネームを持ち、.ovf ファイルの兄弟とする。マニフェストファイルがある場合は、OVF パッケージのコンシューマは、実際の SHA-1 ダイジェストを計算し、その結果をマニフェストファイルに記載されているダイジェストと比べることによって、ダイジェストを検証するものとする。

次のシンタックス定義は、附属書 A に記載する例外を除き、ABNF を使用する。

マニフェストのフォーマットは次の通りである。

```
  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
```

例 2：次の例はマニフェストファイルのコンテンツの一部である。

```
  SHA1(package.ovf)= 237de026fb285b85528901da058475e56034da95
  SHA1(vmdisk1.vmdk)= 393a66df214e192ffbfedb78528b5be75cc9e1c3
```

OVF パッケージはマニフェストファイルに署名することによって署名されることができる。マニフェストファイルのダイジェストは拡張子.cert を持つ証明書ファイルに、base64 エンコード X.509 証明書とともに、保存される。.cert ファイルは.ovf ファイルと同じベースネームをもち、.ovf ファイルの兄弟ファイルとする。OVF パッケージのコンシューマは署名を検証し、証明書を検証すべきである。証明書ファイルのフォーマットは次のようなものとする。

```
  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
```

例 3：次のファイルリストは署名された OVF パッケージの例である。

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

例 4：次の例は見本の OVF 証明書ファイルのコンテンツを示し、マニフェストファイルの SHA1 ダイジェストは 512 ビットキーで署名されている。

```
SHA1(package.mf)= 7f4b8efb8fe20c06df1db68281a63f1b088e19dbf00e5af9db5e8e3e319de
7019db88a3bc699bab6ccd9e09171e21e88ee20b5255cec3fc28350613b2c529089
-----BEGIN CERTIFICATE-----
MIIBgjCCASwCAQQwDQYJKoZIhvcNAQEEBQAwODELMAkGA1UEBhMCQVUxDDAKBgNV
BAgTA1FMRDEbMBkGA1UEAxMSU1NMZWF5L3JzYSB0ZXN0IENBMB4XDTk1MTAwOTIz
MzIwNVoXDTk4MDcwNTIzMzIwNVowYDELMAkGA1UEBhMCQVUxDDAKBgNVBAgTA1FM
RDEZMBcGA1UEChMQTWluY29tIFB0eS4gTHRkLjELMAkGA1UECxMCQ1MxGzAZBgNV
BAMTElNTTGVheSBkZW1vIHNlcnZlcjBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQC3
LCXcScWua0PFLkHBLm2VejqpA1F4RQ8q0VjRiPafjx/Z/aWH3ipdMVvuJGa/wFXb
/nDFLDlfWp+oCPwhBtVPAgMBAAEwDQYJKoZIhvcNAQEEBQADQQArNFsihWIjBzb0
DcsU0BvL2bvSwJrPEqFlkDq3F4M6EgutL9axEcANWgbbEdAvNJD1dmEmoWny27Pn
Ims6ZOZB
-----END CERTIFICATE-----
```

マニフェストファイルと証明書ファイルがある場合は、これらのファイルは OVF 記述子（7.1 参照）の Reference セクションに含まないものとする。これによって OVF 記述子のコンテンツが、OVF パッケージがマニフェストを持っているか、あるいは署名されているかに依存しないことが確認でき、マニフェストまたは証明書をパッケージに追加する決定を、後の段階まで延期することができる。

OVFパッケージの別ファイルが、パッケージマニフェストまたは証明書として解釈される兄弟URLまたはパスネームを占有していなければ、ファイル拡張子.mfと.certをそれらの別ファイルに使用することができる。

5.2 仮想ディスクフォーマット

OVF は特定のディスクフォーマットを使用することを求めていないが、本仕様に準拠するためには、ディスクフォーマットを解釈する方法に関し、制約を受けない仕様を特定する統一資源識別子（URI）によってディスクフォーマットを定めるものとする。仕様は機械が読み取れる形式である必要はないが、ディスクフォーマットをユニークに決定するために、OVF パッケージを読み取るソフトウェアが URI をキーとして使用できるよう、仕様は静的でユニークであるものとする。仕様は十分な情報を提供し、当業者がディスクデータの読み書きどちらもでき、ディスクフォーマットを適切に解釈できるようにするものとする。これらの URI は解決可能であることを推奨する。

5.3 シングルファイルとしての配布

OVF パッケージは TAR フォーマットを使用してシングルファイルとして保管することもある。このファイルの拡張子は.ova（オープン仮想化アプライアンスまたはアプリケーション）とする。

例：次の例はこの種類の OVF パッケージのファイル名の見本を示す。

D:¥virtualappliances¥myapp.ova

シングルファイルとして保存される OVF パッケージでは、OVF 記述子のすべてのファイルリファレンスは相対パスリファレンスであり、TAR アーカイブに含まれるファイルまでの道筋を表すものとする。アーカイブ内の相対ディレクトリは許可されるが、相対パスリファレンスは「..」のドットセグメントを含まないものとする。

TAR 抽出ツールは本来、アーカイブの別部分を修正することなく置き換えるファイルを追加することができるため、要求したファイルをすぐに発見した場合でもアーカイブ全体を読みとらなくてはならないものであった。OVF TAR ファイルでは、アーカイブ内の複製は許可されない。さらに、ファイルは次の順番でアーカイブ内に置かれるものとする。

1）OVF 記述子
2）OVF マニフェスト（任意）
3）OVF 証明書（任意）
4）残りのファイルは Reference セクションに記載するものと同じ順番とする（7.1 参照）。国際化を目的とした外部のストリングリソースバンドルファイルは Reference セクションの最初に記載するものであることに注意する（10 項参照）。
5）OVF マニフェスト（任意）
6）OVF 証明書（任意）

証明書ファイルが任意であることに注意する。証明書ファイルがない場合は、マニフェストファイルも任意となる。マニフェストファイルまたは証明書ファイルがある場合は、どちらも OVF 記述子の後ろに配置されるか、またはどちらもアーカイブの最後に配置されるものとする。

デプロイメントに際しては、順序が決まっているので、OVF 記述子を OVF TAR ファイルから抽出することができ、アーカイブ全体を読み取る必要がない。作成に際しては、順序が決まっているので、OVF TAR ファイルをオンザフライで簡単に作成することができる。順序の制約があっても、標準 TAR パッケージ化ツールを使用して OVF TAR ファイルを作成することができる。

使用する TAR フォーマットはポジックス（POSIX）米国電気電子技術者協会（IEEE）1003.1 標準グループが定義する、USTAR（統一標準テープアーカイブ）に準拠するものとする。

5.4 ファイルセットとしての配布

OVF パッケージはファイルセットとして、標準ウェブサーバ上等で、利用可能である。

例：ウェブサーバ上のファイルセットとしての OVF パッケージの一例を次に挙げる。

```
http://mywebsite/virtualappliances/package.ovf
http://mywebsite/virtualappliances/vmdisk1.vmdk
http://mywebsite/virtualappliances/vmdisk2.vmdk
http://mywebsite/virtualappliances/resource.iso
http://mywebsite/virtualappliances/de-DE-resources.xml
```

6. OVF 記述子

パッケージに関するすべてのメタデータとそのコンテンツは OVF 記述子に保存されている。OVF 記述子は、製品詳細、仮想ハードウェア要件、およびライセンシング等の情報をエンコードする拡張可能な XML 文書である。

OVF 記述子の dsp8023_1.1.0.xsd XML スキーマ定義ファイルはエレメントと属性を含んでいる。

7 項、8 項、9 項は OVF 記述子のセマンティクス、構造、拡張性フレームワークを記載する。これらの項はスキーマ定義読み取りの代わりではなく、スキーマ定義を補足するものである。

OVF 記述子の XML 文書は、トップレベルで許容される唯一のエレメントである 1 つの Envelope エレメントを含むものとする。

本仕様で使用する XML ネームスペースを表 1 に記載する。ネームスペースプレフィクスの選択は自由であり、意味的に重要ではない。

表 1－XML ネームスペースプレフィクス

プレフィクス	XML ネームスペース
ovf	http://schemas.dmtf.org/ovf/envelope/1
ovfenv	http://schemas.dmtf.org/ovf/environment/1
rasd	http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData
vssd	http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData
ovf	http://schemas.dmtf.org/ovf/envelope/1
cim	http://schemas.dmtf.org/wbem/wscim/1/common

7 エンベロープエレメント

Envelope エレメントは（仮想ハードウェアを含む）仮想マシンのすべてのメタデータと、OVF パッケージ自体の構造を記述する。

エンベロープの最も外側のレベルは次のパートから構成される。

- XML ネームスペース URI が定義する、バージョンの表示。
- Reference エレメントとその File の子エレメントが定義する、OVF パッケージの一部であるすべての外部ファイルに対するリファレンスリスト。これらは通常は仮想ディスクファイル、ISO イメージおよび国際化リソースである。
- 9 項で定義するように、セクションエレメントが定義するメタデータ部分。
- シングル仮想マシン（VirtualSystem エレメント）または複数の仮想マシンコレクション（VirtualSystemCollection エレメント）のいずれかの、コンテンツの記述。
- 各ロケールの Strings エレメントが定義する、0 以上のメッセージリソースバンドルの仕様。

例：トップレベルの Envelope エレメントを持つ OVF 記述子構造の例を次に挙げる。

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_VirtualSystemSettingData"
xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
xmlns="http://schemas.dmtf.org/ovf/envelope/1"
xml:lang="en-US"> <References>
<File ovf:id="de-DE-resources.xml" ovf:size="15240"
ovf:href="http://mywebsite/virtualappliances/de-DE-resources.xml"/>
<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> <!-- Describes meta-information about all virtual disks in the package
-->
<DiskSection>
<Info>Describes the set of virtual disks</Info> <!-- Additional section content
-->
</DiskSection>
<!-- Describes all networks used in the package -->
<NetworkSection>
<Info>List of logical networks used in the package</Info> <!-- Additional section
content -->
</NetworkSection>
<SomeSection ovf:required="false">
<Info>A plain-text description of the content</Info>
<!-- Additional section content -->
Version 1.1.0 DMTF Standard 15 </SomeSection>
<!-- Additional sections can follow -->
<VirtualSystemCollection ovf:id="Some Product">
```

```
<!-- Additional sections including VirtualSystem or VirtualSystemCollection-->
</VirtualSystemCollection >
<Strings xml:lang="de-DE">
<!-- Specification of message resource bundles for de-DE locale -->
</Strings>
</Envelope>
```

Envelope エレメントの任意の xml:lang 属性は、記述子内のメッセージを目的とするデフォルトなロケールを特定するものとする。任意の Strings エレメントは様々なロケールのストリングリソースバンドルを含むものとする。国際化サポートに関する詳細は 10 項に述べる。

7.1 ファイルリファレンス

Reference エレメントが定義するファイルリファレンスパートによって、ツールは、記述子の構造全体の解関や解釈をせずに、OVF パッケージのインテグリティを簡単に決定できる。ツールはファイルを失うかもしれないというリスクなしに、OVF パッケージを安全に操作（例えば、コピーやアーカイブ化）できる。

国際化を目的としたストリングリソースバンドルファイルは、最初に Reference エレメントに置くものとする。詳細に関しては、10 項に記載する。

リファレンスパートの各 File エレメントは ovf:id 属性を使用した識別子を与えられる。識別子は OVF パッケージ内ではユニークであるものとする。各 File エレメントは ovf:href 属性を使用して特定するものとし、1 つの URL を含む。相対パスリファレンスと URL スキーマ「file」、「http」、「https」はサポートされるものとする。RFC1738 と RFC3986 を参照すること。その他の URL スキーマを使うべきではない。URL スキーマが特定されていない場合は、ovf:href の属性の値を、OVF 識別子自体の場所に相対するリファレンスファイルのパスネームとして解釈するものとする。相対パスネームは RFC3986 の相対パスリファレンスのシンタックスとして使うものとする。リファレンスファイルは存在すべきである。2 つの異なる File エレメントは、その File エレメントの ovf:href 属性を持つ同一のファイルを参照しないものとする。

リファレンスファイルのサイズを ovf:size 属性を使用して特定することができる。この属性のユニットは常にバイトである。ovf:size 属性が存在するときは、その値はリファレンスファイルの実際のサイズと一致するものとある。

File エレメントが参照する各ファイルは gzip を使用して圧縮することができる（RFC1952 参照）。gzip を使用して File エレメントが圧縮するときは、ovf:compression 属性を「gzip」に設定するものとする。そうしないと、ovf:compression が「identity」に設定されるか、すべての属性が削除される。または、href が HTTP または HTTPS URL ならば、HTTP ヘッダーContent-Encoding: gzip を使用して、HTTP サーバが圧縮を指定できることもある（RFC2616 参照）。ovf:compression 属性と組み合わせて、HTTP コンテンツエンコーディングを使用することが許可されているが、これは一般的には圧縮率を改善しない。圧縮するときには、ovf:size 属性で、実際に圧縮されるファイルのサイズを特定するものとする。

リファレンスパートから参照されたファイルは、任意のファイルシステム上のファイルサイズの制限に合うように、細かくチャンクに分割することもある。チャンクに分割されたことは ovf:chunkSize 属性の存在で表示されるものとする。ovf:chunkSize の値は各チャンクのサイズとするが、最終チャンクはサイズが小さくなることもある。

ovf:chunkSize が特定されているときは、File エレメントはファイル全体のチャンクを代表するチャンクファイルを参照する。この場合は、ovf:href 属性の値は URL の一部のみを特定するものであり、チャンクファイルを分解する URL のシンタックスは次のようになる。附属書 A 記載の例外を除き、シンタックスは ABNF を使用する。

```
chunk-url = href-value "." chunk-number
chunk-number = 9(decimal-digit)
decimal-digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

このシンタックスでは、href の値は ovf:href 属性の値であり、チャンク数は 0 から数えたチャンクの位置であり、値は 0 から始まり、チャンクごとに 1 ずつ増加する。

チャンクは圧縮と組み合わせることができる。まず、全体のファイルをチャンク分割の前に圧縮してから、圧縮ファイルを当分して各チャンクに分割するが、最終チャンクはサイズが小さくなることもある。

OVF パッケージがマニフェストファイルを持つときは、ファイルが複数のチャンクに分割されている場合を除き、マニフェストエントリのファイルネームはファイルの ovf:href 属性の値に一致するものとする。ファイルが複数のチャンクに分割されている場合は chunk-url を使用するものとし、マニフェストファイルは個々のチャンクそれぞれのエントリも含むものとする。チャンクに分割されたファイルでは、マニフェストファイルはファイル全体のエントリを含むことが許可される。その場合は、このダイジェストも検証され

るものとする。

例 1：次の例は様々なタイプのファイルリファレンスを示す。

<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="http://mywebsite/resources/image2.iso"/>

例 2：次の例は前記ファイルリファレンスに対応するマニフェストエントリを示す。

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 コンテンツエレメント

OVF パッケージ内の仮想マシンのコンフィギュレーションは VirtualSystem または VirtualSystemCollection エレメントで表される。これらのエレメントは ovf:id 属性を使用した識別子を与えられるものとする。VirtualSystemCollection の直接の子エレメントはユニークな識別子を持つものとする。

OVF スキーマでは、VirtualSystem および VirtualSystemCollection エレメントは、置換グループのヘッドとして Content エレメントを持つ、置換グループの一部である。Content エレメントは抽象的であり、直接使用することはできない。OVF 記述子は 1 以上の Content エレメントを持つものとする。

VirtualSystem エレメントはシングル仮想マシンを記述し、単にセクションエレメントのコンテナである。 これらのセクションエレメントは仮想ハードウェア、リソース、製品情報を記述し、8 項および 9 項で詳細に説明する。

VirtualSystem エレメントの構造は次の通りである。

```
<VirtualSystem ovf:id="simple-app">
<Info>A virtual machine</Info>
<Name>Simple Appliance</Name>
```

```
<SomeSection>
<!-- Additional section content -->
</SomeSection>
<!-- Additional sections can follow -->
</VirtualSystem>
```

VirtualSystemCollection エレメントは複数の VirtualSystem または VirtualSystemCollection エレメントのコンテナである。従って、任意の複合型コンフィギュレーションを記述することができる。VirtualSystemCollection レベルのセクションエレメントはアプライアンスの情報、プロパティ、リソース要件などを記述し、9 項で詳細に説明する。

VirtualSystemCollection エレメントの構造は次の通りである。

```
<VirtualSystemCollection ovf:id="multi-tier-app">
<Info>A collection of virtual machines</Info>
<Name>Multi-tiered Appliance</Name>
<SomeSection>
<!-- Additional section content -->
</SomeSection>
<!-- Additional sections can follow -->
<VirtualSystem ovf:id="...">
<!-- Additional sections -->
</VirtualSystem>
<!-- Additional VirtualSystem or VirtualSystemCollection elements can follow-->
</VirtualSystemCollection>
```

Content 置換グループのすべてのエレメントは、1 つの Info エレメントを含むものとし、1 つの Name エレメントを含むこともある。Info エレメントには、このエンティティの意味を人間が読める形で記述したものが含まれる。Name エレメントは、コンテンツのローカライズ可能な任意のディスプレイネームである。Info と Name エレメントをローカライズする方法に関する詳細は 10 項に記載する。

7.3 拡張性

本仕様を使用することによって、いくつかの方法でカスタムメタデータを OVF 記述に追加

できる。

- 新しいセクションのエレメントは Section 置換グループの一部として定義され、OVF スキーマがセクションの存在を許可する場所で使用される。Section のすべてのサブタイプには、このエンティティの意味を人間が読める形で記述する Info エレメントが含まれる。Info エレメントの値は、例えばセクションが抜かされたときに、パーサがセクションについて何も知らない場合でも、ユーザに重要な警告を出すように使用される。Info エレメントをローカライズする方法に関する詳細は 10 項に記載する。
- OVF スキーマは、既存のすべてのタイプが追加エレメントなしに拡張できるオープンコンテンツモデルを使用する。拡張ポイントは namespace="##other"を持つ xs:any 宣言と共に OVF スキーマで宣言される。
- OVF スキーマによって、既存のタイプに追加の属性が許可される。

カスタム拡張は本仕様で定義する XML ネームスペースを使用しないものとする。これはカスタムエレメントとカスタム属性の両方に適用する。

カスタムエレメントでは、ブーリアン ovf:required 属性によって、エレメントの情報が正しい動作のために必要であるか、または任意であるかが特定される。特定されていない場合は、ovf:required 属性のデフォルトは TRUE となる。必要とされる拡張子および理解できない拡張子を検知する OVF パッケージのコンシューマは機能しないものとする。

既知の Section エレメントでは、理解されていない追加の子エレメントが発見され、その ovf:required 属性値が TRUE である場合は、OVF パッケージのコンシューマは、理解しない 1 つのものとしてセクション全体を解釈するものとする。チェックは帰納的ではなく、Section エレメントの直接の子エレメントにのみ適用される。

この動作によって古いパーサは、そうしないように明確な指示がない限り、新しい OVF 仕様を確実に拒否できる。

カスタム属性上では、属性内の情報は正しい動作を必要とはしないものとする。
例 1：

```
<!-- Optional custom section example -->
<otherns:IncidentTrackingSection ovf:required="false">
<Info>Specifies information useful for incident tracking purposes</Info>
<BuildSystem>Acme Corporation Official Build System</BuildSystem>
<BuildNumber>102876</BuildNumber>
```

```
<BuildDate>10-10-2008</BuildDate>
</otherns:IncidentTrackingSection>
```

例 2 :

```
<!-- Open content example (extension of existing type) -->
<AnnotationSection>
<Info>Specifies an annotation for this virtual machine</Info>
<Annotation>This is an example of how a future element (Author) can still be
parsed by older clients</Annotation>
<!-- AnnotationSection extended with Author element -->
<otherns:Author ovf:required="false">John Smith</otherns:Author>
</AnnotationSection>
```

例 3 :

```
<!-- Optional custom attribute example -->
<Network ovf:name="VM network" otherns:desiredCapacity="1 Gbit/s">
<Description>The main network for VMs</Description>
</Network>
```

7.4 適合性

本仕様は OVF 記述子の 3 つの適合性レベルを定義し、1 が適合性レベルの最上位である。

- OVF 記述子は本仕様で定義するセクション、エレメント、属性のみを使用する。
  適合性レベル : 1
- OVF 記述子は本仕様で定義されていないカスタムセクションまたはエレメントまたは属性を使用し、その拡張子は 7.3 で定義するように任意である。
  適合性レベル : 2
- OVF 記述子は本仕様で定義していないカスタムセクションまたはエレメントを使用し、少なくともその 1 つの拡張子が 7.3 で定義するように必要とされている。必要とされるすべての拡張子の定義は、オープンで制約を受けない XML スキーマにおいて広く利用可能であるものとする。完全な仕様は XML スキーマに内包されることもあり、また別の文書として利用されることもある。
  適合性レベル : 3

適合性レベル 3 の使用は可搬性を制限するため、可能な限り避けるべきである。

適合レベルは OVF 記述子に直接特定されていないが、前術の規則に従って決定されるものとする。

8 仮想ハードウェア記述子

8.1 VirtualHardwareSection

VirtualSystem の各エレメントは 1 以上の VirtualHardwareSection エレメントを含むことがあり、そのエレメントそれぞれが、仮想システムが必要とする仮想ハードウェアを記述する。仮想マシンが必要とする仮想ハードウェアは、VirtualHardwareSection エレメントで特定される。本仕様は、主要なデバイスのみが記述されている抽象的または不完全なハードウェア記述をサポートする。ハイパーバイザは、記述子に記載された必要なデバイスが実現できるのであれば、追加の仮想ハードウェアコントローラおよびデバイスを作成することができる。

この仮想ハードウェア記述は CIM クラス CIM_VirtualSystemSettingData と CIM_ResourceAllocationSettingData に基づく。CIM モデルの XML 表示は WS-CIM マッピングに基づく（DSP0230）。

例：VirtualHardwareSection の例：

```
<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>
```

```
<!-- Additional Item elements can follow -->
</VirtualHardwareSection>
```

VirtualSystem エレメントは VirtualHardwareSection の直接の子エレメントを持つものとする。VirtualHardwareSection は VirtualSystemCollection エレメントおよび Envelope エレメントの直接の子エレメントとして許可されない。

シングル VirtualSystem エレメント内での、複数の VirtualHardwareSection エレメントの存在は許可されている。OVF パッケージのコンシューマは、特定の仮想化プラットフォームに最適な仮想ハードウェアを選択すべきである。VirtualHardwareSection エレメントはエレメントを特定するために使用できる ovf:id 属性を含むこともある。ovf:id 属性がある場合は、その値は VirtualSystem 内でユニークでなくてはならない。

ovf:transport 属性は、OVF 環境文書内で仮想マシンにプロパティを届けるトランスポートメカニズムのタイプを特定する。この属性は、ゲスト/プラットフォームコミュニケーションメカニズムを提供することを目的とする、プラグ可能かつ拡張可能なアーキテクチャをサポートする。いくつかのトランスポートタイプをスペースで区切って、特定することができる。9.5 にプロパティの説明を、11 項にトランスポートタイプと OVF 環境の説明を記載する。

vssd:VirtualSystemType エレメントは、仮想システムのタイプをユニークに識別する実装依存のストリングである、仮想システムタイプ識別子を特定する。例えば、VMware の第 4 世代仮想ハードウェアに対する仮想システムタイプ識別子は vmx-4 となり、Xen の第三世代仮想ハードウェアに対する仮想システムタイプ識別子は xen-3 となる。ゼロ以上の仮想システムタイプ識別子はシングルスペースで区切って別々に特定されることもある。OVF 仮想システムをターゲットプラットフォーム上でデプロイするためには、ターゲットプラットフォーム上の仮想マシンは、vssd:VirtualSystemType エレメントが特定する少なくとも 1 つの仮想システムタイプをサポートすべきである。vssd:VirtualSystemType エレメントで特定される仮想システムタイプ識別子は、CIM クラス CIM_VirtualSystemManagementCapabilities の VirtualSystemTypesSupported のプロパティの値と一致するものと考えられる。

仮想ハードウェアの特徴は Item エレメントのシークエンスとして記述される。Item エレメントは、CIM クラス CIM_ResourceAllocationSettingData のインスタンスの XML 表現である。エレメントは仮想ハードウェアデバイスに加えて、すべてのメモリおよび CPU 要件を記述できる。

複数のデバイスサブタイプをシングルスペースで区切って、1 つの Item エレメントで特定することができる。
例：

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

8.2 拡張性

Item エレメント上の任意の ovf:required 属性は、エレメントの実現（例えば、CD-ROM または USB コントローラ）が、ゲストソフトウェアの正しい動作に必要かを特定する。特定されていない場合は、ovf:required のデフォルトは TRUE となる。

Item エレメントの子エレメントが異なる RASD WS-CIM ネームスペースにあったとしても、これらの子エレメント上で、任意のブーリアン属性 ovf:required が解釈されるものとする。Item エレメントのパーサツールは表 2 に従って作動すべきである。

表 2－ovf:required 属性を持つ子エレメントの動作

子エレメント	ovf:required 属性の値	行動
既知	TRUE または指定なし	Item を解釈する
既知	FALSE	Item を解釈する
未知	TRUE または指定なし	Item を不合格にする
未知	FALSE	Item を無視する

8.3 仮想ハードウェアエレメント

VirtualHardwareSectionエレメント内のItemエレメントの一般的なフォームは次の通りである。

```
<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>
<!-- multiple connection elements can be specified -->
```

```
<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>
```

エレメントは CIM_ResourceAllocationSettingData クラスによってエクスポーズされるプロパティを表す。それらは DSP1041、特定のリソースタイプに対して DSP1041 から派生したプロファイル、そして本文書で定義する設定のセマンティクスを持つ。
例：次の例はメモリサイズの記述を示す。

```
<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>
```

Description エレメントはエレメント自体の追加メタデータを提供するために使用される。このエレメントによって、OVF パッケージのコンシューマは、アプリケーションが書かれた頃には未知であったアイテムも含めて、すべてのアイテムの記述情報を提供することが

できる。

Caption、Description、ElementName エレメントは、OVF エンベロープネームスペースの ovf:msgid 属性を使用してローカライズすることができる。国際化サポートに関する詳細は 10 項に記載する。

任意の ovf:configuration 属性は、コンフィギュレーションネームのリストを含む。この属性のセマンティクスのデプロイメントオプションについては 9.8 に記載する。任意の ovf:bound 属性を範囲を特定するために使用する。8.4 参照。

ディスク、CD-ROM、ネットワーク等のデバイスはデプロイメントプラットフォームからのバッキングが必要である。バッキングの要件は HostResource または Connection エレメントのいずれかを使用して特定する。

イーサネットアダプタに対しては、Connection エレメントにロジカルネットワークネームを特定する。OVF パッケージ内で同じロジカルネットワークネームを参照する複数のイーサネットアダプタは 同じネットワーク上にデプロイするものとする。

HostResource エレメントは、デプロイメントプラットフォーム上のロジカルデバイスと、OVF 記述子内のリソースを参照するために使用される。OVF 記述子に含まれるリソースを参照する HostResource エレメントの値は表 3 で特定する URI としてフォーマットされる。

表 3－HostResource エレメント

コンテンツ	記述
ovf:/file/<id>	リファレンスセクションで特定される、OVF 内のファイルのリファレンス。<id>はリファレンスされている File エレメントの ovf:id 属性の値とする。
ovf:/disk/<id>	DiskSection で特定される、仮想ディスクへのリファレンス。<id>はリファレンスされている Disk エレメントの ovf:diskId 属性の値とする。

バッキングを必要とするデバイスにバッキングが特定されていない場合は、デプロイメントプラットフォームは、例えば、ユーザをプロンプトする等、適切な選択をするものとする。1 以上のバッキングを 1 台のデバイスに特定することは許可されていない。

表 4 は仮想デバイスおよびコントローラを記述するために、エレメントを使用する方法に関する概要を示す。

表 4－仮想デバイスおよびコントローラのエレメント

エレメント	使用
rasd:Description	人間が読める形での、情報の意味の記述。例えば、「仮想マシンのメモリサイズを特定する。」
rasd:ElementName	人間が読める形での、コンテンツの記述。例えば、「256MB メモリ。」
rasd:InstanceID	セクション内のエレメントの、ユニークなインスタンス ID。
rasd:HostResource	抽象的に、デプロイメントプラットフォーム上でデバイスをリソースに接続する方法を特定する。すべてのデバイスがバッキンを必要とするわけではない。表 3 参照。
rasd:ResourceType rasd:OtherResourceType rasd:ResourceSubtype	記述されているデバイスの種類を特定する。
rasd:AutomaticAllocation	フロッピー、CD-ROM、イーサネットアダプタ等の接続可能なデバイスに対して、このエレメントは、デバイスの電源がオンの状態で接続すべきかを特定する。
rasd:Parent	親コントローラ（があるとき）、そのインスタンスID。
rasd:Connection	イーサネットアダプタに対して、このエレメントは仮想マシンの抽象的なネットワーク接続名を特定する。OVF パッケージ内で同じ抽象的なネットワーク接続名を特定するイーサネットアダプタは、同じネットワーク上でデプロイされるものとする。抽象的なネットワーク接続名は最も外側のエンベロープレベルでNetworkSection に記載されるものとする。
rasd:Address	デバイス特有。イーサネットアダプタでは、このエレメントはMACアドレスを特定する。
rasd:AddressOnParent	デバイスでは、このエレメントはコントローラ上の位置を特定する。
rasd:AllocationUnits	使用されるアロケーションユニットを特定する。例えば、例えば「byte * 2^20。」
rasd:VirtualQuantity	示されているリソースの量を特定する。例えば、「256。」

rasd:Reservation	利用可能を保証できるリソースの最小量を特定する。
rasd:Limit	保証されたリソースの最大量を特定する。
rasd:Weight	その他の割り当てに関連してこの割り当ての相対優先度を特定する。

デバイスの記述に直接関連するフィールドのみ言及する。すべてのフィールドの完全な記述については、CIF MOF を参照のこと。各フィールドは CIM_ResourceAllocationSettingData クラスの同一の名前を持つプロパティに対応する。

8.4 エレメント上の範囲

任意の ovf:bound 属性を使用して、Item エレメントの範囲を特定するために使用することができる。範囲は最小、標準、最大値を持ち、min、normal、max と示し、min <= normal <= max とする。mix と max のデフォルト値は normal として特定する値である。

OVF パッケージをデプロイするプラットフォームは標準値で開始し、継続的なパフォーマンス調整と検証の範囲内で値を調整することを推奨する。

VirtualHardwareSection と ResourceAllocationSection エレメント内の Item エレメントでは、次の追加セマンティクスが定義される。

- 各 Item エレメントは任意の ovf:bound 属性を持つ。この値は min、max、または normal として特定される。デフォルト値は normal である。属性が特定されていないか、または normal と特定されている場合は、アイテムは標準の仮想ハードウェアまたはリソースアロケーション記述の一部と解釈される。
- ovf:bound 値が min または max のいずれかとして特定されている場合は、アイテムは 1 以上の任意の InstanceID の上限または下限を特定するために使用される。このようなアイテムはレンジマーカーと呼ばれる。

レンジマーカーのセマンティクスは次の通りである。

- InstanceID と ResourceType を特定し、ResourceType は同じ InstanceID を持つ別の Item エレメントと一致するものとする。
- 1 以上の min レンジマーカーまたは 1 以上の max レンジマーカーを 1 つの任意の RASD （InstanceID で識別される）に対して特定することは無効である。
- レンジマーカーを持つ、ある Item エレメントは、レンジマーカーを持たない対応する Item エレメントを持つものとする。つまり、ovf:bound 属性を持たない Item エレメン

ト、または ovf:bound 属性を持ち、標準値である Item エレメントを意味する。この対応するアイテムはデフォルト値を特定する。

- min 範囲のみ特定されている Item の max 値は、プロパティの有効値内で無制限に高い値である。
- max 範囲のみ特定されている Item の min 値は、プロパティの有効値内で無制限に低い値である。
- デフォルト値はレンジ内にあるものとする。
- レンジマーカーRASD 内での非整数エレメントの使用は無効である。

例：次の例はレンジマーカーの使用例を示す。

```
<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">
<rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
<rasd:ElementName>1024 MB maximum memory size</rasd:ElementName>
<rasd:InstanceID>0</rasd:InstanceID>
<rasd:Reservation>1024</rasd:Reservation>
<rasd:ResourceType>4</rasd:ResourceType>
```

```
</Item>
</VirtualHardwareSection>
```

9 コアメタデータセクション

表 5 は定義されるコアメタデータを示す。

表 5—コアメタデータセクション

セクション	ロケーション	多重度
DiskSection パッケージ内の仮想ディスクのメタ情報を記述する。	エンベロープ	ゼロまたは 1
NetworkSection パッケージ内で使用するロジカルネットワークを記述する。	エンベロープ	ゼロまたは 1
ResourceAllocationSection 仮想マシンコレクションの任意のリソースである、メモリまたは CPU の予約、リミット、シェアを特定する。	VirtualSystemCollection	ゼロまたは 1
AnnotationSection エンティティの自由形式のアノテーションを特定する。	VirtualSystem VirtualSystemCollection	ゼロまたは 1
ProductSection 製品名、バージョン等のパッケージの製品情報を、設定できるプロパティとともに特定する。	VirtualSystem VirtualSystemCollection	ゼロ以上
EulaSection パッケージ内のソフトウェアのライセンス契約を特定する。	VirtualSystem VirtualSystemCollection	ゼロ以上
StartupSection 仮想マシンコレクションの電源をオンにする方法を特定する。	VirtualSystemCollection	ゼロまたは 1
DeploymentOptionSection 目標とするリソース要件を具体的に特定する。	エンベロープ	ゼロまたは 1

OperatingSystemSection 仮想マシンにインストールしたゲストオペレーティングシステムを特定する。	VirtualSystem	ゼロまたは 1
InstallSection 仮想マシンにソフトウェアをインストールし設定するために、最初に仮想マシンを起動する必要があることを特定する。	VirtualSystem	ゼロまたは 1

次のサブクローズでは、コアセクションのセマンティクスを説明し、例をいくつか示す。このセクションは OVF エンベロープの複数の場所で使用される。各セクションの記述によって、どこで使用できるかを定義する。すべての属性とエレメントの詳細な仕様に関しては OVF スキーマを参照のこと。

OVF スキーマでは、すべてのセクションは置換グループのヘッドとして Section エレメントを持つ、置換グループの一部である。Section エレメントは抽象的であり、直接使用することはできない。

9.1 DiskSection

DiskSection は OVF パッケージの仮想ディスクに関するメタ情報を記述する。仮想ディスクとそのメタデータは仮想ハードウェアの外部に記述され、OVF パッケージ内の仮想マシン間の共有を促進する。

例：次の例は仮想ディスクの記述を示す。

```
<DiskSection>
<Info>Describes the set of virtual disks</Info>
<Disk ovf:diskId="vmdisk1" ovf:fileRef="file1" ovf:capacity="8589934592"
ovf:populatedSize="3549324972"
ovf:format=
"http://www.vmware.com/interfaces/specifications/vmdk.html#sparse">
</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 はエンベロープの最も外側のレベルでのみ有効なセクションである。

各仮想ディスクは ovf:diskId 属性を使用する識別子を与えられた Disk エレメントで表され、識別子は DiskSection 内でユニークであるものとする。

仮想ディスクのキャパシティは xs:long 整数値を持つ ovf:capacity 属性によって特定されるものとする。アロケーションのデフォルトユニットはバイトとする。任意のストリング属性 ovf:capacityAllocationUnits を使用して、特定のアロケーションユニットを特定することができる。ovf:capacityAllocationUnits の値は、ベースユニットを「バイト」とする制限を持ち、DSP0004 で定義するプログラムユニットのフォーマットに一致するものとする。

ovf:fileRef 属性は、Reference エレメント内の既存の File エレメントを認識することで、仮想ディスクコンテンツを表示し、File エレメントはその ovf:id 属性値と ovf:fileRef 属性値を一致させることによって識別される。ovf:fileRef 属性を削除することは、ディスクが空であることを意味するものとする。この場合はディスクを作成し、ディスク全体のコンテンツがインストール時にはゼロであるとする。ゲストソフトウェアは通常、エンプティディスクを何らかのシステムフォーマットにフォーマットする。

空でない仮想ディスクのフォーマット URI（5.2 参照）は、ovf:format 属性で特定されるものとする。

異なる Disk エレメントは同一の値を持つ ovf:fileRef 属性を含まないものとする。Disk エレメントは、Reference エレメントで定義した順序と同じ順序で File エレメントが特定できるような順序とする。

空のディスクに対して固定した仮想ディスクキャパシティを特定するのではなく、空のディスクのキャパシティを OVF プロパティを使用して、ovf:capacity="${disk.size}のように決めることもできる。OVF プロパティは xs:long 整数値に分解する。OVF プロパティの記述については 9.5 に記載する。ユーザがプロンプトされて、ディスクサイズ情報を、例えばギガバイトで入力することもできるため、ovf:capacityAllocationUnits 属性は、OVF プロパティを使用する際に役立つ。

空ではないディスクでは、実際に使用されているディスクのサイズを ovf:populatedSize 属性を使用して任意に特定できる。この属性のユニットは常にバイトである。

ovf:populatedSize は使用されているディスクサイズの推定として認められるが、ovf:capacity より大きくないものとする。

VirtualHardwareSection では、仮想ディスクデバイスは DiskSection 内の Disk エレメントを参照する rasd:HostResource エレメントを持つこともある（8.3 参照）。仮想ディスクキャパシティは Disk エレメント上の ovf:capacity 属性によって定義されるものとする。rasd:VirtualQuantity エレメントが rasd:HostResource エレメントと共に特定される場合は、仮想量の値を考慮しないものとし、どの値でも持つことができる。

OVF によって、ディスクイメージを親イメージと比較して、修正ブロックセットとして提示することができる。親ディスクを使うことによって、OVF パッケージが類似するコンテンツを持つ複数のディスクを含む場合は、OVF パッケージのサイズを大幅に減少することができる。Disk エレメントに関しては、ovf:parentRef 属性を使って親ディスクを任意に特定することができ、異なる Disk エレメントを参照する有効な ovf:diskId リファレンスを含むものとする。ディスクブロックがローカルに存在しない場合は、そのディスクブロックを検査し、親ディスクに発生する。DiskSection では、親 Disk エレメントを参照する子 Disk エレメントより前に、親 Disk エレメントが発生するものとする。

9.2 NetworkSection

NetworkSection エレメントは OVF パッケージで使用するすべてのロジカルネットワークをリストするものとする。

```
 <NetworkSection>
<Info>List of logical networks used in the package</Info>
<Network ovf:name="red">
<Description>The network the Red service is available on</Description>
</Network>
<Network ovf:name="blue">
<Description>The network the Blue service is available on</Description>
</Network>
</NetworkSection>
```

NetworkSection はエンベロープの最も外側のレベルで有効なエレメントである。

すべての VirtualHardwareSection エレメント内の Connection エレメントから参照される

ネットワークは NetworkSection で定義するものとする。

9.3 ResourceAllocationSection

ResourceAllocationSection エレメントは VirtualSystemCollection エンティティのすべてのリソースアロケーション要件を記述する。これらのリソースアロケーションは OVF パッケージをデプロイするときに遂行するものとする。

```
<ResourceAllocationSection>
<Info>Defines reservations for CPU and memory for the collection of VMs</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 は VirtualSystemCollection エンティティの有効なエレメントである。

任意の ovf:configuration 属性はコンフィギュレーションネームのリストを含む。この属性のセマンティクスのデプロイメントオプションに関しては 9.8 に記載する。

任意の ovf:bound 属性は min、max、または normal の値を含む。この属性の意味については 8.4 に記載する。

9.4 AnnotationSection

AnnotationSection エレメントはエンティティに対してユーザが定義するアノテーションである。これらのアノテーションは OVF パッケージをデプロイする際に表示されることがある。

```
<AnnotationSection>
<Info>An annotation on this service. It can be ignored</Info>
<Annotation>Contact customer support if you have any problems</Annotation>
</AnnotationSection >
```

AnnotationSection は VirtualSystem および VirtualSystemCollection エンティティの有効なエレメントである。

Annotation エレメントのローカライズ方法に関する詳細については、10 項に記載する。

9.5 ProductSection

ProductSection エレメントは、製品名、バージョン、ベンダ等のアプライアンスの製品情報を特定する。

```
<ProductSection ovf:class="com.mycrm.myservice" ovf:instance="1">
<Info>Describes product information for the service</Info>
<Product>MyCRM Enterprise</Product>
<Vendor>MyCRM Corporation</Vendor>
<Version>4.5</Version>
<FullVersion>4.5-b4523</FullVersion>
<ProductUrl>http://www.mycrm.com/enterprise</ProductUrl>
<VendorUrl>http://www.mycrm.com</VendorUrl>
<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 address of administrator</Description>
</Property>
<Category>Admin properties</Category>
<Property ovf:key="app.log" ovf:type="string" ovf:value="low"
```

```
ovf:userConfigurable="true">
<Description>Loglevel for the service</Description>
</Property>
<Property ovf:key="app.isSecondary" ovf:value="false" ovf:type="boolean">
<Description>Cluster setup for application server</Description>
</Property>
<Property ovf:key="app.ip" ovf:type="string" ovf:value="${appserver-vm}">
<Description>IP address of the application server VM</Description>
</Property>
</ProductSection>
```

任意の Product エレメントは製品名を特定し、任意の Vendor エレメントは製品ベンダ名を特定する。任意の Version エレメントは製品のバージョンを短縮形で特定し、任意の FullVersion エレメントは製品のバージョンを省略しないで記述する。任意の ProductUrl エレメントは、人間が読める形の製品の記述に解決される URL を特定し、任意の VendorUrl は人間が読める形のベンダの記述に解決される URL を特定する。

任意の AppUrl エレメントはデプロイされた製品インスタンスに解決される URL を特定する。このエレメントは実験的である。任意の Icon エレメントは製品のディスプレイアイコンを特定する。このエレメントは実験的である。

Property エレメントはアプリケーションレベルのカスタマイズされたパラメータを特定し、ネットワークアイデンティティ、DNS サーバの IP アドレス、ゲートウェイ等の、特定の設定を持つデプロイメント中にカスタマイズを必要とするアプライアンスに、特に関する。

ProductSection は VirtualSystem および VirtualSystemCollection エンティティの有効なセクションである。

Property エレメントは Category エレメントを使用してグループ分けできる。Category エレメントによってグループ化された Property エレメントの集まりは、Category エレメントに続く Property エレメントのシークエンスであるが、Property エレメントではないエレメントの前まで続く。大量の Property エレメントを含む OVF パッケージは、これを使用することで、より単純なインストールを体験できる。同様に、各 Property エレメントは、Description 子エレメントによって定義される記述に加えて、Label 子エレメントによって定義される短いラベルを持つこともある。Category エレメントと、Property エレメントの

Description と Label 子エレメントのローカライズ方法に関する詳細については 10 項に記載する。

ProductSection の各 Property エレメントは ovf:key 属性を使用して、ProductSection 内でユニークな識別子を与えられる。

ProductSection の各 Property エレメントは ovf:type 属性を使用してタイプを与えられ、任意に ovf:qualifiers 属性を使用してタイプ修飾子を与えられる。有効なタイプを表 6 に記載し、有効な修飾子を表 7 に記載する。

任意の属性 ovf:value は、プロパティのデフォルト値を提供するために使用される。1 以上の任意の Value エレメントが、9.8 で定義される特別なコンフィギュレーションの代替デフォルト値を定義するために使用されることもある。

任意の属性 ovf:userConfigurable はインストール段階でのプロパティ値が、コンフィギュレーション可能であるか決定する。ovf:userConfigurable が FALSE または削除されている場合は、ovf:value 属性はインストール中のカスタマイズパラメータに使用する値を特定する。ovf:userConfigurable が TRUE の場合は、ovf:value 属性はそのカスタマイズパラメータのデフォルト値を特定するが、インストール中に変更されることもある。

コマンドラインのインストーラーのようなシンプル OVF の実装は、通常プロパティのデフォルト値を使用し、ovf:userConfigurable が TRUE に設定されていても、プロンプトしない。設定時にプロンプトを強制するためには、整数タイプでは空のストリングは有効な整数値ではないため、ovf:value 属性を削除すれば十分である。ストリングタイプには、空ではないストリングを要求する修飾子を追加することによってプロンプトを強制することができる。表 7 を参照のこと。

任意のブーリアン属性 ovf:password はプロパティ値が機密情報を含むことを示す。デフォルト値は FALSE である。プロパティ値をプロンプトする OVF 実装は、ovf:password が TRUE に設定されているときに、それらの値を覆い隠すよう忠告を受ける。これは HTML の password タイプのテキスト入力と同様である。このメカニズムは限られたセキュリティ保護しか提供できないことに注意する。敏感な値は無関心な観察者からは隠されているが、OVF 記述子のデフォルト値および OVF 環境の割り当てられた値は、まだクリアテキストでみることができるからである。

ゼロ以上の ProductSections は VirtualSystem または VirtualSystemCollection 内で特定

することもできる。通常は、ProductSection はインストールされている特定のソフトウェア製品に対応する。同じエンティティレベルの各製品セクションはユニークな ovf:class と ovf:instance 属性をペアで持つ。シングル ProductSection だけが使用される一般的な場合には、ovf:class および ovf:instance 属性は任意であり、空のストリングにはデフォルトである。リバースドメイン名規約を使用するソフトウェア製品をユニークに特定するために、ovf:class プロパティを使用することを推奨する。値の例は com.vmware.tools および org.apache.tomcat である。同一製品の複数のインスタンスがインストールされている場合は、ovf:instance 属性を使用して、様々なインスタンスを特定する。

OVF 環境を通じてゲストソフトウェアにプロパティエレメントがエクスポーズされる。11 項記載のように、OVF 環境内でエクスポーズされる Property エレメントの ovfenv:key 属性の値は、次のように OVF 記述子の ProductSection のエンティティ内で定義される、対応する Property エレメントの ovf:key 属性の値から構築するものとする。

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

ここで、

- class-value は ProductSection エンティティで定義する Property エレメントの ovf:class 属性の値である。製品 [class-value "."]は class-value が空のストリングでない場合にのみ存在するものとする。
- key-value-prod は ProductSection エンティティで定義する Property エレメントの ovf:key 属性の値である。
- instance-value は ProductSection エンティティで定義する Property エレメントの ovf:instance 属性の値である。製品 [instance-value "."]は instance-value が空のストリングでない場合にのみ存在するものとする

例 : 次の OVF 環境例はプロパティをゲストソフトウェアにプロパゲートする方法について示す。

```
<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"/>
```

OVF パッケージのコンシューマは ovf:userConfigurable が TRUE のとき、プロパティをプロンプトすべきである。これらのプロパティは OVF パッケージのサブエンティティに加えて、複数の ProductSections においても定義されることもある。

ProductSection が存在する場合は、パッケージのトップレベルの Content エレメントで定義される最初の ProductSection エンティティが、パッケージ全体を記述する要約情報を定義するものとする。インストール後、OVF パッケージのコンシューマはこの情報を CIM_Product クラスのインスタンスとして利用できるようにすることができる。

VirtualSystemCollection で特定する Property エレメントも、直接の子によって見られる。（11 項参照）。子は親 VirtualSystemCollection のプロパティに、ovf:value 属性の値として${name}のフォーム上のマクロスを使用して参照することもできる。

表 6 はプロパティの有効なタイプを記載する。これらは DSP0004 で定義する CIM 固有のタイプのサブセットであり、DSP0004 は各固有タイプの値スペースおよびフォーマットも定義する。各 Property エレメントは ovf:type 属性を使用してタイプを特定するものとする。

表 6－プロパティタイプ

タイプ	記述
uint8	符号なし 8 ビット整数
sint8	符号付き 8 ビット整数
uint16	符号なし 16 ビット整数
sint16	符号付き 16 ビット整数
uint32	符号なし 32 ビット整数
sint32	符号付き 32 ビット整数
uint64	符号なし 64 ビット整数
sint64	符号付き 64 ビット整数
string	ストリング
boolean	ブーリアン
real32	IEEE 4 バイト浮動小数点
real64	IEEE 8 バイト浮動小数点

表 7 は DSP0004 で定義する、サポートされている CIM タイプの修飾子を記載する。各 Property エレメントは ovf:qualifiers 属性を使用して、コンマで区切られた複数の修飾子と共に、任意にタイプを特定するものとする。附属書 A の DSP0004 の「MOF シンタックスグラマー記述」の qualifierList を参照のこと。

表 7－プロパティ修飾子

タイプ	記述

string	MinLen(min) MaxLen(max) ValueMap{...}
uint8 sint8 uint16 sint16 uint32 sint32 uint64 sint64	ValueMap{...}

9.6 EulaSection

EulaSection には、その親 Content エレメントを使用するための法的用語が含まれる。このライセンスは OVF パッケージのデプロイメント中に表示され、承認されるものとする。複数の EulaSections が OVF に存在することもある。無人のインストールが許可される場合は、すべての埋め込みライセンスセクションが暗示的に承認される。

```
<EulaSection>
<Info>Licensing agreement</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>
```

EulaSection は VirtualSystem および VirtualSystemCollection エンティティの有効なセクションである。

License エレメントをローカライズする方法の詳細については、10 項に記載する。

9.7 StartupSection

StartupSection は仮想マシンコレクションの電源を、オンおよびオフにする方法を特定する。

```
<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>
```

VirtualSystemCollection の直接の子である各 Content エレメントは、VirtualSystemCollection エンティティの StartupSection エンティティにおいて、対応する Item エレメントを持つこともある。Item エレメントが、VirtualSystem および VirtualSystemCollection エンティティの両方に対応できることに注意する。開始または終了動作が VirtualSystemCollection エンティティ上で行われる場合は、StartupSection エンティティの Item エレメント上の関連動作が特定の順序で呼び出される。Item エレメントが（ネストされた）VirtualSystemCollection エンティティに対応するときはいつも、StartupSection の Item エレメント上の動作が、VirtualSystemCollection エンティティに対応するアイテムエレメントが呼び出される前に、呼び出されるものとする（つまり、深さ優先トラバーサル）

次の Item 上で要求される属性は、VirtualSystem および VirtualSystemCollection のためにサポートされている。

- ovf:id はこの VirtualSystemCollection の直接の子である Content エレメントの ovf:id 属性の値と一致するものとする。この Content エレメントは、Item エレメントで定義されている動作が適用される仮想マシンまたは仮想マシンコレクションを記述する。
- ovf:order は非負の整数値を使用して開始オーダーを特定する。開始動作の実行オーダーは値の数値昇順である。同じオーダー識別子を持つ Item は同時に開始されることもある。停止動作の実行オーダーは値の数値降順である。

次の Item 上の任意属性は VirtualSystem のためにサポートされている。

- ovf:startDelay は開始シークエンスの次の順序に移る前の待機時間を秒単位で特定する。デフォルト値は 0 である。
- ovf:waitingForGuest によって、ゲストソフトウェアがその準備が整ったことを報告した後に、プラットフォームは開始シークエンスを再開することができる。この解釈はデプロイメントプラットフォーム特有である。デフォルト値は FALSE である。
- ovf:startAction は使用する開始動作を特定する。有効な値は powerOn と none である。デフォルト値は powerOn である。
- ovf:stopDelay は停止動作の前の順序に進む前に待機する時間を秒単位で特定する。デフォルト値は 0 である。
- ovf:stopAction は使用する停止動作を特定する。有効値は powerOff、guestShutdown および none である。guestShutdown の解釈はデプロイメントプラットフォーム特有である。デフォルト値は powerOff である。

特定されていない場合は、コレクション内の各エンティティに ovf:order="0"で黙示的デフォルトアイテムが作成される。従って、自明な開始シークエンスに関しては、StartupSection を特定する必要はない。

9.8 DeploymentOptionSection

DeploymentOptionSection は目的とするコンフィギュレーションの離散集合を特定する。OVF パッケージの作者は様々なコンフィギュレーションのサイジングメタデータを含むこともできる。OVF のコンシューマは、例えばユーザをプロンプトすることでコンフィギュレーションを選択するものとする。選択されたコンフィギュレーションは OVF 環境において可視であるため、ゲストソフトウェアは選択されたコンフィギュレーションに適応することができる。11 項参照。

DeploymentOptionSection は各コンフィギュレーションの ID、ラベル、記述を特定する。

```
<DeploymentOptionSection>
<Configuration ovf:id="Minimal">
<Label>Minimal</Label>
<Description>Some description</Description>
</Configuration>
<Configuration ovf:id="Typical" ovf:default="true">
<Label>Typical</Label>
<Description>Some description</Description>
</Configuration>
```

```
<!-- Additional configurations -->
</DeploymentOptionSection>
```

DeploymentOptionSection は次のセマンティクスを持つ。

- DeploymentOptionSection がある場合は、エンベロープレベルでのみ有効であり、OVF 記述子には 1 つのセクションのみが特定されるものとする。
- コンフィギュレーションの離散集合は、Configuration エレメントと共に記述され、パッケージ内でユニークである、ovf:id 属性によって特定される識別子を持つものとする。
- デフォルトの Configuration エレメントは任意の ovf:default 属性と共に特定することができる。デフォルトが特定されていない場合は、リストの最初のエレメントがデフォルトである。1 以上のエレメントをデフォルトとして特定することは無効である。
- Label および Description エレメントは ovf:msgid 属性を使用してローカライズできる。国際化サポートに関する詳細は 10 項に記載する。

仮想ハードウェアおよび仮想マシンコレクションのリソースを管理するためにコンフィギュレーションを使用することができる。VirtualHardwareSection エレメントの Item エレメントは、VirtualSystem エンティティのリソースを記述し、ResourceAllocationSection エレメントの Item エレメントは仮想マシンコレクションのリソースを記述する。これらの 2 つの Item タイプに対して、次の追加セマンティクスを定義する。

- 各アイテムは任意の ovf:configuration 属性を持ち、シングルスペースで区切られたコンフィギュレーションのリストを含む。特定されていない場合は、どのコンフィギュレーションからでもアイテムを選択するものとする。特定されている場合は、アイテムは選択したコンフィギュレーションの ID がリストにある場合のみ、アイテムを選択するものとする。コンフィギュレーション属性は DeploymentOptionSection に特定されていない ID を含まないものとする。
- シングル VirtualHardwareSection または ResourceAllocationSection 内では、複数の Item エレメントが、同じインスタンス ID を参照することを許可される。任意のインスタンス ID のシングルの複合 Item は各 Item エレメントの子エレメントをピックアップすることで構築されるものとし、OVF 記述子における前出の Item エレメントの子エレメントは、後者の Item エレメントに似た名前の子エレメントがある場合は、選ばない。Item エレメントの子エレメントに特定される属性で、上記の方法で選ばれなかったものは、複合 Item エレメントの一部ではない。
- すべての Item エレメントはリソースタイプを特定するものとし、同じインスタンス ID を持つ Item エレメントはリソースタイプで一致するものとする。

例 1：次の例は VirtualHardwareSection を示す。

```
 <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>
```

Item 上の属性 ovf:configuration および ovf:bound は、非常に柔軟なコンフィギュレーションのオプションを提供するために、組み合わせて使用することもできる。

プロパティのデフォルト値を管理するために、コンフィギュレーションをさらに使用することもできる。ProductSection 内の Property エレメントに対して、次の追加セマンティクスを定義する。

- DeploymentOptionSection の様々なコンフィギュレーションに対して、代替的なデフォルト値を用いることができる。Label および Description エレメントに加えて、各 Property エレメントは任意に Value エレメントを含むことができる。Value エレメントは代替デフォルトを特定する ovf:value 属性と、この新しいデフォルト値を使うべき

コンフィギュレーションを特定する ovf:configuration 属性を持つものとする。複数の Value エレメントは同じコンフィギュレーションを参照しないものとする。

例 2：次に ProductSection の例を示す。

```
 <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 OperatingSystemSection

OperatingSystemSection は仮想マシンにインストールされたオペレーティングシステムを特定する。

```
<OperatingSystemSection ovf:id="76">
<Info>Specifies the operating system installed</Info>
<Description>Microsoft Windows Server 2008</Description>
</OperatingSystemSection>
```

ovf:id の有効値は CIM_OperatingSystem.OsType プロパティの ValueMap 修飾子で定義される。

OperatingSystemSection は VirtualSystem エンティティのみに有効なセクションである。

9.10 InstallSection

InstallSection が特定されている場合は、ゲストソフトウェアをインストールおよび/または設定するために、仮想マシンを一度起動する必要があることを示す。ゲストソフトウェアは OVF の起動中に OVF 環境にアクセスし、ソフトウェアのインストールおよび/または設定終了後、ゲストの電源をオフにしてシャットダウンすることを期待されている。

InstallSection が特定されていない場合は、仮想マシンがゲストソフトウェアのインスールを完了するために、電源をオンにする必要がないことを示す。

```
<InstallSection ovf:initialBootStopDelay="300">
<Info>Specifies that the virtual machine needs to be booted once after having created
the guest software in order to install and/or configure the software
</Info>
</InstallSection>
```

InstallSection は VirtualSystem エンティティのみに有効なセクションである。

任意の ovf:initialBootStopDelay 属性は仮想マシンの電源をオフにするまでの待機時間を秒で特定する。設定されていない場合は、実装には、仮想マシン自体が電源オフになるまで待つものとする。待機時間が終了後も仮想マシンの電源がオフにならない場合は、OVF パッケージのコンシューマはエラーを表示するものとする。

仮想マシンのゲストソフトウェアは電源をオフにする前に、複数の再起動を行うことがあることに注意する。

仮想マシンコレクションの複数の VM が定義された InstallSection を持つことがあり、その場合は、前述の段階は VM ごとに同時に行われることが多い。

10 国際化

任意の ovf:msgid 属性を使用して、次のエレメントはローカライズ可能なメッセージをサポートする。

- Content 上の Info エレメント
- Content 上の Name エレメント
- Section 上の Info エレメント
- AnnotationSection 上の Annotation エレメント
- EulaSection 上の License エレメント
- NetworkSection 上の Description エレメント
- OperatingSystemSection 上の Description エレメント
- ProductSection 上の Description、Product、Vendor、Label、Category エレメント
- Property 上の Description と Label エレメント
- DeploymentOptionSection 上の Description と Label エレメント
- VirtualHardwareSection における System エレメント上の ElementName、Caption、Description サブエレメント
- VirtualHardwareSection における Item エレメント上の ElementName、Caption、Description サブエレメント

- ResourceAllocationSection における Item エレメント上の ElementName、Caption、Description サブエレメント

ovf:msgid 属性は、様々なロケールで、様々な値を持つこともあるメッセージを参照する識別子を含む。
例 1：

```
<Info ovf:msgid="info.text">Default info.text value if no locale is set or no locale match</Info>
<License ovf:msgid="license.tomcat-6_0"/> <!-- No default message -->
```

エンベロープエレメント上の xml:lang 属性は、記述子内のメッセージのデフォルトロケールを特定するものとする。属性は任意であり、「en-US」のデフォルト値を持つ。

メッセージリソースバンドルは OVF 記述子の内側でも外側でも良い。内部のリソースバンドルは Envelope エレメントの末端の Strings エレメントとして表される。
例 2：

```
<ovf:Envelope xml:lang="en-US">
...
... sections and content here ...
...
<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>
```

外部リソースバンドルは References セクションにまず記載されるものとし、Strings エレメントから参照される。外部メッセージバンドルは埋め込みメッセージバンドルと同じスキーマを取る。外部メッセージバンドルには正確に 1 つの Strings エレメントが存在し、その Strings エレメントは特定の ovf:fileRef 属性を持たないこともある。
例 3：

```
<ovf:Envelope xml:lang="en-US">
<References>
...
<File ovf:id="it-it-resources" ovf:href="resources/it-it-bundle.msg"/>
```

```
</References>
... sections and content here ...
...
<Strings xml:lang="it-IT" ovf:fileRef="it-it-resources"/> ... </ovf:Envelope>
```

例 4： 前例で参照されている、外部リソース/ it-it-bundle.msg file のコンテンツ例。

```
<Strings
xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
xmlns="http://schemas.dmtf.org/ovf/envelope/1"
xml:lang="it-IT">
<Msg ovf:msgid="info.os">Sistema operativo</Msg>
...
</Strings>
```

内部および外部 Strings エレメントは交互に配置することができるが、Envelope エレメントの末端に配置するものとする。任意のロケールを持つ msg:id 属性が複数生じる場合は、後者が前者を上書きする。

11 OVF 環境

OVF 環境はゲストソフトウェアとデプロイメントプラットフォームがやりとりする方法を定義する。この環境によって、ゲストソフトウェアは、OVF 記述子に定義されるプロパティのユーザ特有値等、デプロイメントプラットフォームに関する情報にアクセスすることができる。

環境の仕様はプロトコル部分とトランスポート部分に分割される。プロトコル部分は、ゲストソフトウェアがアクセスできる XML 文書のフォーマットとセマンティクスを定義する。トランスポート部分はデプロイメントプラットフォームとゲストソフトウェア間で情報を交換する方法を定義する。

OVF 環境の dsp8027_1.1.0.xsd XML スキーマ定義ファイルはエレメントと属性を含む。

11.1 環境文書

環境文書は拡張 XML 文書であり、ゲストソフトウェアが実行される環境に関する情報をゲストソフトウェアに提供する。文書を得る方法はトランスポートタイプによって異なる。
例：OVF 環境文書の構造例は次の通りである。

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovfenv="http://schemas.dmtf.org/ovf/environment/1"
xmlns="http://schemas.dmtf.org/ovf/environment/1"
ovfenv:id="identification of VM from OVF descriptor">
<!-- Information about virtualization platform -->
<PlatformSection>
<Kind>Type of virtualization platform</Kind>
<Version>Version of virtualization platform</Version>
<Vendor>Vendor of virtualization platform</Vendor>
<Locale>Language and country code</Locale>
<TimeZone>Current timezone offset in minutes from UTC</TimeZone>
</PlatformSection>
<!--- Properties defined for this virtual machine -->
<PropertySection>
<Property ovfenv:key="key" ovfenv:value="value">
<!-- More properties -->
</PropertySection>
<Entity ovfenv:id="id of sibling virtual system or virtual system collection">
<PropertySection>
<!-- Properties from sibling -->
</PropertySection>
</Entity> </Environment>
```

Environment エレメントの ovfenv:id 属性の値はこの仮想マシンを記述する VirtualSystem エンティティの ovf:id 属性の値と一致するものとする。

PlatformSection エレメントはデプロイメントプラットフォームが提供する任意の情報を含む。Kind、Version、Vendor エレメントはデプロイメントプラットフォームベンダの詳細を記述する。これらのエレメントは実験的である。Locale と TimeZone エレメントは現在のロケールとタイムゾーンを記述する。これらのエレメントは実験的である。

PropertySection は Property エレメントを含み、現在の仮想マシンの OVF 記述子で特定されるすべてのプロパティに対応するキー/値のペアを持ち、直接の親

VirtualSystemCollection があれば、その特定のプロパティも持つ。環境はプロパティをシンプルリストとして表し、アプリケーションが構文解析しやすくする。さらに、シングルリストフォーマットは VirtualSystem 上のプロパティが、親 VirtualSystemCollection 上に定義されるプロパティに優先する、優先セマンティクスをサポートする。優先されたプロパティはリストに載らないものとする。優先は、現在の仮想マシンのプロパティと、親 VirtualSystemCollection のプロパティが同一の ovf:key、ovf:class、ovf:instance 属性値を持つときに生じる。9.5 参照。この場合には、優先された親プロパティの値は、マクロを持つ親プロパティを参照する、異なる名前のついた子プロパティを加えることによって得ることができる。9.5 参照。

兄弟 VirtualSystem と VirtualSystemCollection がある場合は、それぞれに対して 1 つの Entity エレメントが存在するものとする。Entity エレメントの ovfenv:id 属性の値は、兄弟エンティティの ovf:id 属性の値に一致するものとする。Entity エレメントはキー/価値プロパティのペアを兄弟の OVF 環境文書に持ち、特定の兄弟 Entity エレメントのコンテンツは、その兄弟が見る正確な PropertySection を持つものとする。この情報によって、IP アドレス等のコンフィギュレーション情報を、多層アプリケーションの一部となる VirtualSystems が利用できるようになる。

図 8 は定義されているコアセクションを示す。

表 8－コアセクション

セクション	ロケーション	多層性
PlatformSection デプロイメントプラットフォームからの情報を提供する。	環境	ゼロまたは1
PropertySection OVF記述子で定義されるプロパティに対応するキー/値のペアを含む。	環境 エンティティ	ゼロまたは1

環境文書は新しいセクションタイプを提供することによって拡張可能である。文書のコンシューマは未知のセクションタイプとエレメントを無視すべきである。

11.2 トランスポート

環境文書情報はゲストソフトウェアと多くの方法で通信することができる。これらの方法はトランスポートタイプと呼ばれる。トランスポートタイプは OVF 記述子に

VirtualHardwareSection の ovf:transport 属性として特定される。シングルスペースで区切られた、いくつかのトランスポートタイプを特定することができ、この場合はどれでも自由に実装できる。トランスポートタイプは環境文書がデプロイメントプラットフォームからゲストソフトウェアへ通信される方法を特定する。

相互運用可能性を実現するために、この仕様は CD-ROM デバイスをサポートするすべての実装がサポートする必要がある、「iso」トランスポートタイプを特定する。iso トランスポートは、ゲストソフトウェアが利用可能となる、動的に作成された ISO イメージを作ることによって環境文書と通信する。iso トランスポートタイプをサポートするため、仮想マシンを起動する前に、実装によって ISO 読み取り専用ディスクイメージを DM-ROM 切断に備えてバッキングとして作るものとする。iso トランスポートが VirtualHardwareSection に選択される場合は、少なくとも 1 つの切断された CD-ROM デバイスがこのセクション内に存在するものとする。

作成された ISO イメージは Joilet 拡張子をサポートする ISO9660 仕様に順守するものとする。

ISO イメージはこの特定の仮想マシンに対する環境文書を持つものとし、環境は ISO イメージのルートダイレクトリ内に含まれる ovf-env.xml と名づけられた XML ファイル内に存在するものとする。ゲストソフトウェアは今標準ゲストオペレーティングシステムツールを使用して情報にアクセスすることができる。

起動前に仮想マシンが 1 以上の切断された CD-ROM を持っていた場合は、ゲストソフトウェアは ovf-env.xml ファイルを持つ ISO イメージの位置をつかむために、接続した CD-ROM デバイスをスキャンしなくてはならないこともある。

OVF 環境を持つ ISO イメージは、仮想マシンを起動するたびにゲストソフトウェアに利用可能なものとする。

「iso」トランスポートタイプのサポートは、CD-ROM デバイスサポートを持たない仮想ハードウェアアーキテクチャまたはゲストオペレーティングシステムに必要ではない。

本仕様を準拠するために、URI が iso 以外のトランスポートフォーマットを提供するものとし、トランスポートの使用法に関し、制約のない仕様を特定する。仕様は機械読み取り形式でなくてもよいが、OVF 記述子を読むソフトウェアが、フォーマットをユニークに決定する際にキーとして使用できるように、静的でユニークであるものとする。仕様は当業者

が、プロトコルを実装するトランスポートメカニズムを解釈するためには十分であるものとする。これらの URI は解決可能であることを推奨する。

附属書A
（情報提供）
シンボルと規約

XML例は表1で定義するネームスペースプレフィクスを使用する。XML例は子エレメント上のネームスペースプレフィクスを特定しないスタイルを使用する。XML規則は、ネームスペースプレフィクスを持たない特定された子エレメントは親エレメントから派生し、XML文書のデフォルトネームスペースから派生するものではないと定義することに注意する。文書全体を通して、XMLエレメント値の余白は読みやすさのために使用される。実際には、サービスは余白が使われていないかのように、エレメント値内の余白前後を許容しストリップする。

拡大BNF (ABNF)におけるシンタックス定義はIETF RFC5234が定義するABNFを次の例外を除き使用する。

- ABNFで定義するスラッシュ（/）の代わりに、バー（|）で区切られた規則は選択を意味する。
- ABNFで定義するように大文字小文字を区別しないのではなく、すべての文字は大文字と小文字を区別して処理しなければならない。
- ABNFで定義するようにエレメントを余白抜きでアセンブルする代わりに、余白（つまりスペース字U+0020およびタブ字U+0009）はシンタックスエレメント間で許容される。

附属書B
（情報提供）
変更履歴

バージョン	日付	記述
1.0.0	2009年2月22日	DMTF標準
1.1.0	2010年1月12日	DMTF標準

附属書C
（情報提供）
OVF XSD

本仕様の XML スキーマの規範的コピーは次の URL を参照して入手できる。

http://schemas.dmtf.org/ovf/envelope/1/dsp8023_1.1.0.xsd

http://schemas.dmtf.org/ovf/environment/1/dsp8027_1.1.0.xsd

本仕様の XML スキーマの xs:documentation コンテンツは情報を提供するものであり、便宜を図るためのみに提供される。

CIM_ResourceAllocationSystemSettingsData および CIM_VirtualSystemSettingData の WS-CIM マッピング（DSP0230）の XML スキーマの規範的コピーは次の URL を参照して入手することができる。

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2.22.0/CIM_VirtualSystemSettingData.xsd

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2.22.0/CIM_ResourceAllocationSettingData.xsd

本仕様は次の CIM MOF に基づく。

参考文献

ISO 9660, *Joliet Extensions Specification*, May 1995,
http://bmrc.berkeley.edu/people/chaffee/jolspec.html

W3C, Y. Savourel et al, *Best Practices for XML Internationalization*, Working Draft, October 2007,
http://www.w3.org/TR/2007/WD-xml-i18n-bp-20071031

DMTF DSP1044, *Processor Device Resource Virtualization Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1044_1.0.pdf

DMTF DSP1045, *Memory Resource Virtualization Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1045_1.0.pdf

DMTF DSP1047, *Storage Resource Virtualization Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1047_1.0.pdf

DMTF DSP1022, *CPU Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1022_1.0.pdf

DMTF DSP1026, *System Memory Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1026_1.0.pdf

DMTF DSP1014, *Ethernet Port Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1014_1.0.pdf
DSP1050, *Ethernet Port Resource Virtualization Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1050_1.0.pdf