



## オープン仮想化フォーマット・ホワイトペーパー

バージョン**1.0.0**

位置付け：情報提供

公開日：2009年2月6日

**DSP2017**

Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit <http://www.dmtf.org/about/policies/disclosures.php>.

DMTF は、企業やシステムの管理、および相互運用性の推進に力を注いでいる業界のメンバーから成る非営利団体である。メンバー、およびメンバー以外でも、所属を正しく示すことを条件に、DMTF の仕様と文書を複製することができる。DMTF の仕様は時折改定されることがあるため、バージョンおよび公開日には常に注意を払うべきである。

本標準または標準案の特定のエレメントを実装することは、仮特許権を含む第三者の特許

権（本書では「特許権」という）の対象となることもある。DMTF は標準のユーザーに対し、前記権利の存在について何ら表明するものではなく、前記第三者の特許権、特許権者、主張者のいずれかまたはすべてを認識、公開、または特定する責任を負わない。また、前記権利、特許権者、主張者の不完全または不正確な特定、公開に対しても責任を負わない。DMTF はいかなる相手に対しても、いかなる方法または環境、またいかなる法論理においても、前記の第三者特許権を認識、公開、または特定しなかったことに対し何ら責任を負わず、前記第三者の標準に関する信頼性、またはその製品、プロトコル、試験方法論に組み込まれた標準に関する信頼性に対しても何ら責任を負わないものとする。DMTF は、前記標準の実装が予測できるか否かにかかわらず、前記標準を実装するいかなる相手に対しても、またいかなる特許権者または主張者に対しても、何ら責任を負わないものとする。また DMTF は、公開後に標準が撤回または修正されることによって生じるコストや損失に対し何ら責任を負わず、また、標準を実装するいかなる相手からも、前記実装に対する特許権者の侵害のあらゆる主張に対して、何ら損害を受けず、免責されるものとする。

第三者が保有する特許権であって、DMTF 標準の実装に関連するか、または影響を与える可能性があるかと特許権者が考え、すでに DMTF に通知済みである特許権に関する情報については、次のサイトで参照できる。<http://www.dmtf.org/about/policies/disclosures.php>.

#### 要約

本ホワイトペーパーはオープン仮想化フォーマット（OVF）を説明する。OVF は1つ以上の仮想コンピューター・システムから成る仮想アプライアンスのパッケージ化と配布を目的とした、ハイパーバイザー非依存で、効率性、拡張性に優れた、オープンな仕様である。本ホワイトペーパーが対象とする読者は、OVF と OVF の開発の理由を理解したいと希望する者全員である。仮想化および CIM モデルの一般概念に精通していることが前提となる。

## 内容

1 はじめに	6
1.1 概要	6
1.2 仮想アプライアンス	7
1.3 設計の目標	8
1.4 仮想アプライアンスのライフサイクル	11
2 ポータビリティのある仮想化フォーマット	12
2.1 OVF パッケージ	12
2.2 OVF 環境	13
2.3 OVF 記述子の見本	14
3 オープン仮想化フォーマットの使用	17
3.1 作成	17
3.2 デプロイメント	18
4 特徴	20
4.1 仮想ハードウェア記述	20
4.2 デプロイメント・オプション	22
4.3 デプロイメントのカスタマイズ	24
4.4 国際化	25
4.5 拡張可能性	26
4.6 適合性	27
5 ポータビリティ	27
6 OVF 仕様の将来のバージョン	30
7 結論	30
A 多層 Petstore 例	32
アーキテクチャーとパッケージ化	32
プロパティ	33
ウェブゲスト・ソフトウェア	33
データベースゲスト・ソフトウェア	34
ディスクのレイアウト	35
完全な OVF 記述子	36
完全な OVF 環境	47
バージョン 1.0.0	4 ページ

Web 層仮想マシンの OVF 環境	47
B LAMP スタック例	49
デプロイメント時のカスタマイズ	50
シンプル LAMP OVF 記述子	52
2 層 LAMP OVF 記述子	58
C 拡張性の例	68
カスタム・スキーマ	69
カスタム拡張を持つ記述子	70

## 1 はじめに

### 1.1 概要

仮想化インフラストラクチャーの導入が急速に進んだことで、仮想化プラットフォームに対して、あるいは仮想化プラットフォーム間で仮想マシンを配置するための、標準的でポータビリティのあるメタデータ・モデルの必要性が浮かび上がって来た。オペレーティング・システムと、そのオペレーティング・システム上で認証されるアプリケーションをパッケージ化し、独立系ソフトウェア・ベンダー（ISV）から仮想マシンとして簡単に移管され、試験と開発を経て、外部環境に依存しない、構成済み・パッケージ済みのユニットとして製品化することは、非常に魅力的である。このような仮想マシン（VM）としてパッケージ化され、事前にデプロイされ、すぐに動作できるアプリケーションは仮想アプライアンスと呼ばれる。この概念を大規模に実現するには、どの仮想化プラットフォーム上でも仮想アプライアンスを自動的かつ安全にインストール、設定、動作させるために必要となる、上記の VM とメタデータをパッケージ化するベンダー非依存の標準を業界で導入することが重要となる。

仮想アプライアンスによって、アプリケーション・ビルダはアプリケーション用にソフトウェア・スタックを最適化することができ、ターンキー・ソフトウェア・サービスをエンドユーザーに提供できるため、仮想アプライアンスはソフトウェア配布のパラダイムを変えつつある。ソリューション・プロバイダーにとっては、仮想アプライアンスを構築する方が、ハードウェア・アプライアンスを構築するよりも簡単でコスト効率が良い。なぜなら、アプリケーションは使用するオペレーティング・システムとパッケージ済みであるため、ISV はアプリケーションと OS の整合性テストと認証を省略し、ソフトウェアが動作する OS 環境内にソフトウェアを事前導入することができるからである。エンドユーザーにとっては、仮想アプライアンスを使うことによって、現在の OS とアプリケーション個別の管理タスクに代わって、標準化、自動化された効率的なプロセスを導入することで、ソフトウェア運用管理ライフサイクルを大幅に単純化する機会を持つことができる。

現在の仮想アプライアンスがシングル VM だけを含むのに対して、現代企業のアプリケーションは、各層が 1 つ以上のマシンを持つ複数階層のサービス指向アーキテクチャー（SOA）をモデルにする。シングル VM モデルは従って、多層サービスの配布には十分ではない。さらに、複合型アプリケーションではネットワークやカスタマ固有のプロパティを、インストール時にカスタマイズする必要がある。その上、仮想アプライアンスは、特定のハイパーバイザーに適したハードディスク・イメージと構成データを持つ実行時フォーマット

にパッケージ化される。実行時フォーマットは実行に最適化されるが、配布には最適化されない。効率的なソフトウェア配布には、ポータビリティ、プラットフォーム独立性、検証、署名、バージョニング、ライセンス条項をはじめとする、その他多くの特徴が非常に重要となる。

オープン仮想化フォーマット (OVF) 仕様は 1 つ以上の VM から成る、仮想アプライアンスのパッケージ化と配布を目的とした、ハイパーバイザー非依存で、効率性、拡張性に優れた、オープンな仕様である。OVF の目的は、仮想マシンだけではなく、機能単位としてのアプライアンスを、自動的かつ安全に管理することである。OVF フォーマットが成功を収めるためには、ISV、仮想アプライアンス・ベンダー、オペレーティング・システム・ベンダー、および仮想化プラットフォーム・ベンダーによる OVF フォーマットの開発、支持が必要であり、その開発は、標準に基づく枠組みの範囲内でなければならない。

本文書では OVF 設計の背景にある動機と目標を詳細に説明する。本書は同じ改定番号を持つ OVF 仕様の付随文書として読まれるべきである。

## 1.2 仮想アプライアンス

仮想アプライアンスは事前構成されたソフトウェア・スタックであり、1 つ以上の仮想マシンから構成される。各仮想マシンは独立してインストールできる実行時エンティティであり、オペレーティング・システム、アプリケーション、その他のアプリケーション固有データ、そして仮想マシンが要求する仮想ハードウェアの仕様から構成される。基盤アプリケーションの多くが仮想アプライアンスとして利用可能であるが、ネットワーク経由でアクセスできる DNS サーバー、バグ追跡データベース、またはウェブ、アプリケーションおよびデータベース層から成る完全な CRM ソリューション等のエンドユーザー・アプリケーションであっても、仮想アプライアンスとして利用可能である。複雑なソフトウェアシステムとサービスを、事前構成されたソフトウェア・スタックとして提供することによって、堅牢性が劇的に向上し、インストールが簡単になる。仮想アプライアンスはサードパーティ ISV によって開発、提供される必要はなく、その概念は企業にとっても同様に使いやすい。特定のサービス用の仮想マシン・テンプレートを企業の IT 組織が組立て、試験して認証し、その後、パッケージ化して企業全体で何度も繰り返せる定型的なデプロイメントを行うために、仮想アプライアンスが使用されることも多い。

一般的にソフトウェア・サービスは、複数の仮想マシンで動作し、ネットワークを介して通信する多層アプリケーションとして実装される。サービスは別のサービスで構成される

ことが多く、その別のサービス自体も多層アプリケーションであるか、また別のサービスで構成されることもある。この概念はサービス指向アーキテクチャ (SOA) として知られている。実際に、仮想アプライアンスはネットワーク経由の、XML ベースの管理およびサービス・インターフェースを使用することが一般的であり、そのサービス・インターフェースによって、完全なアプリケーションを提供するアプライアンスを組み合わせられるため、SOA タイプのモデルは仮想アプライアンス・ベースのインフラストラクチャーに無理なく適合する。

例えば、3 層から成るウェブ・アプリケーションを想定する。プレゼンテーション・ロジックを実装する Web 層と、ビジネス・ロジックを実装するアプリケーション・サーバー層と、バックエンド・データベース層の 3 層から成る。単純な実装では、この 3 層を 1 つにつき 1 層として、3 つの仮想マシンに振り分けることになる。この方法では、アプリケーションを 1 つの物理ホストから 3 つの物理ホストに拡張できる。各層をそれぞれサービスとして扱う方法もある。従って、各層がクラスター化されたソリューションを提供する複数 VM サービスとなる。この方法では、物理ホスト 3 つのみより、はるかに大きな拡張性を持つことができる。ウェブフロントの例を取り上げると、一般的なシナリオでは、ウェブ・サーバー多数と、アプリケーション少数、データベース・サーバーを 1 つか 2 つ持つ。仮想マシンとして実装されれば、各層は必要となる最大数または最小数の物理マシンに対応することができ、各層はサービス VM のインスタンスを複数サポートできる。

### 1.3 設計の目標

オープン仮想化フォーマット (OVF) は、仮想マシン (または仮想マシン・コレクション) 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、ポータビリティ、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特性は次の通りである。

- 配布用に最適化

業界標準の公開鍵インフラストラクチャーに基づき、コンテンツの検証と整合性の確認をサポートし、ソフトウェアライセンスの基本的な管理計画を提供する。

- シンプルで自動化されたユーザー・エクスペリエンスに最適化

VM のライフサイクル管理プロセスのインストール段階における、OVF のパッケージ全体、各仮想マシンまたはメタデータ構成要素の検証をサポートする。OVF はユーザーが読み取れる形のアプライアンスに関連する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



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。

- ・ ポータビリティのある VM パッケージ化

OVF は仮想化プラットフォームに中立的であるが、プラットフォーム特有の機能強化も可能にしている。OVF は現在 VM 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おり、将来開発されるフォーマットにも対応できるように拡張可能である。仮想マシンのプロパティは簡潔、正確に記録される。

- ・ ベンダーおよびプラットフォーム非依存

OVF は（アプライアンスにおける）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・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カスタム・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

複数の地域でユーザー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、対話処理のローカライズをサポートする。この機能によって、パッケージ化されたシングル・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ーの協力から生まれ、ポータビリティのある仮想マシンの将来の標準として、発展していくと見込まれている。

ユーザーの観点から見ると、OVF はソフトウェア・アプライアンスのパッケージ化フォーマットである。一度インストールされると、OVF はユーザーのインフラストラクチャーに、自己完結型で、自律性を持ち、特定の目標を達成することを目的としたソフトウェア・ソリューションを追加する。OVF は例えば、LAMP スタック（Linux + Apache + MySQL + PHP）等、完全に機能する試験済みのウェブ・サーバー/データベース/OS の組み合わせを含むこともあり、また、ウイルス・チェッカーを、その更新ソフトウェア、スパイウェア検知ソフトウェア等をあわせて含むこともある。

技術的な観点から見ると、OVF は仮想マシン・テンプレートの搬送機構である。1つの OVF はシングル VM、または複数の VM を含むこともでき、ソフトウェア・アプライアンス開発者がどちらの構成がアプリケーションに適切であるか決めることができる。OVF はインストールされて初めて、動作できる。特定の仮想化プラットフォームでは OVF から VM を

動作できることもあるが、必須ではない。この場合は、仮想マシンの実行時状態が OVF に浸透するため、OVF 自体をアプライアンスの理想のバージョンとみなすことはできない。さらに、プラットフォームが OVF の整合性を確認するためのデジタル署名も無効となる。

搬送機構として、OVF は VMware の VMDK 仮想ディスク・フォーマットおよび Microsoft の VHD 仮想ディスク・フォーマット、またはオープンソース QCOW フォーマットとは異なる。これらはシングル VM ディスクの範囲内で動作する実行時 VM イメージのフォーマットであり、現在では搬送フォーマットとして使用されることが多いものの、VM のポータビリティに関する問題を解決するように設計されていない。複数のディスクを持つ VM、複数 VM の場合、インストール時に VM のカスタマイズが必要な場合、または複数の仮想化プラットフォーム上で動作させる VM の場合には、(仮想化プラットフォームが特定の仮想化ハードディスク・フォーマットの使用をサポートすると公言している場合であっても)、前記のフォーマットは助けにならない。

OVF に含まれているのは、パッケージ化されたソフトウェア仮想アプライアンスの認証と保全の概念であり、それによってプラットフォームがアプライアンスの出自を決定することができ、エンドユーザーは適切に信頼性を判断することができる。OVF 仕様は、アプライアンス自身がそのコンフィギュレーションと修正に責任を持つように構築されてきた。具体的には、仮想化プラットフォームがアプライアンスのファイルシステムを読み取れなくても良いことを意味する。アプライアンスからプラットフォームを分離することは、OVF をどのオペレーティング・システムを使用しても実装でき、OVF フォーマットをサポートするどの仮想化プラットフォームにもインストールできるということを意味する。アプライアンスは、特定のメカニズムを使用して、インストールされたプラットフォームを検知し、反応する。このため、プラットフォームは業界全体のアプライアンスの相互互換性を損なうことなく、独特な方法で本仕様を拡張することができる。

OVF フォーマットは、複合型多層サービス、およびその配布、インストール、コンフィギュレーションおよびワークフロー実施のために設計された特定の特徴を持つ。

- OVF は複合サービスを提供するために、多層アプリケーションのコンフィギュレーションと仮想マシンの構成を直接サポートする。
- OVF は VM とアプリケーション・レベル両方のコンフィギュレーションの仕様記述を許可する。
- OVF は OVF のコンテンツを検証する強力なメカニズムを提供し、ユーザーによる導

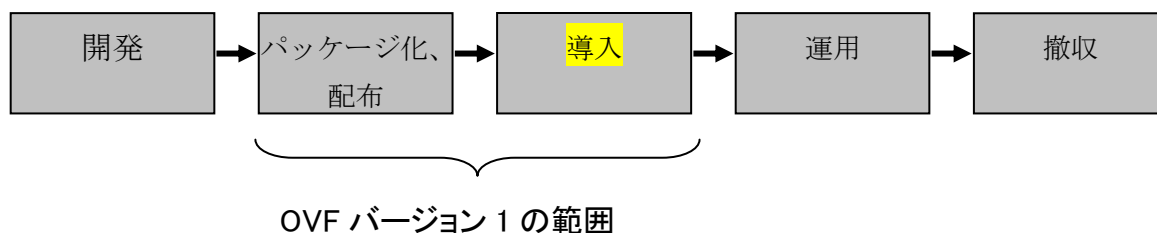
入作業を軽減するために無人インストールを完全にサポートし、結果としてユーザー・エクスペリエンスを強化する。

- ・ OVF は市場で受け入れられている手続きである、署名とトラステッド・サードパーティを使用して、OVF のコンテンツの整合性を確認する。これにより、アプライアンス作成者が署名して以降、アプライアンスが改ざんされていないことを利用者に保証することができる。この点が、仮想アプライアンス市場が成功するために、そしてアプライアンスを独立に作成し、オンラインからダウンロードできるようにするために、非常に重要であると考えられている。

- ・ OVF はアプライアンスに関するライセンス条項を提示して承認を得るための基本方法を提供することで、アプライアンス・ベンダーとユーザーの商業的利益を守る。

#### 1.4 仮想アプライアンスのライフサイクル

仮想アプライアンスのソフトウェアのライフサイクルを次に示す。



1つ以上のVMと関連するコンフィギュレーションおよびデプロイメント・メタデータから成るサービスは、開発最終段階でOVFフォーマットにパッケージ化される。ここで使用される構成要素は第三者のものでもよい。例えば、クラスター化されたデータベース構成要素は第三者ISVからも取得できる。デプロイメント段階はOVFパッケージのインストールを意味する。運用段階と撤収段階は使用する仮想化製品、OVF自体のコンテンツに特有である。運用には、例えば、アプライアンスの継続したメンテナンスとアップグレードが含まれるが、OVFのVMコンテンツに大きく依存することが多い。撤収段階では、ソフトウェアは動作を中止し、ソフトウェアが使用しているリソースは開放される。本OVF仕様バージョンでは、パッケージ化、配布およびデプロイメント段階を具体的に扱う。後に公開されるバージョンでは運用と撤収段階について詳細に扱う予定である。

## 2 ポータビリティのある仮想化フォーマット

オープン仮想化フォーマットは、仮想マシンにデプロイされるソフトウェアの配布フォーマット、および OVF が実行する環境を定義する。

### 2.1 OVF パッケージ

OVF パッケージは OVF 記述子と、通常は仮想ディスクである追加コンテンツのセットから成る。コンテンツは直接パッケージにはいるものもあれば、HTTP 経由で外部参照されるものもある。本仕様により、OVF パッケージ全体を 1 つのファイルとして配布することも可能である。

OVF 記述子は XML 文書であり、仮想ディスクにインストールされたソフトウェアに関するメタデータを記述する。OVF 仕様 1.0 は、効率的にソフトウェアをデプロイするために使用する共通セクションである、仮想ハードウェア、ディスク、ネットワーク、リソース要件、カスタム化パラメータ等を定義する。記述子は拡張可能に設計されているため、後でさらに情報を追加できる。

本仕様では、ディスク・フォーマット仕様が公開されていて制約がない限り、どのような仮想ディスク・フォーマットも使用できる。つまり、現在ハイパーバイザーに使用されている仮想ハードディスク・フォーマットの全範囲をサポートし、将来のフォーマットも使用できるように拡張可能である。

仮想ディスク・フォーマットは一般に、インストールしたゲスト OS を問わず、簡潔なベーシック・ディスク・ブロック・フォーマットとなる。例えば VMware VMDK は、フラット、スパースおよび圧縮の形式で、64KB ブロックに保存される 512 バイト・ディスク・セクタを処理する。デプロイメント時に、仮想化プラットフォームは好ましいベーシック・ディスク・ブロック・フォーマットで仮想ディスクを作成する。実行時仮想ディスク・フォーマットは配布フォーマットと同一でも良いが、違うこともある。例えば、圧縮された仮想ディスク・フォーマットを実行するには効率的ではないかもしれない。最後に、仮想ディスクにインストールされたゲスト OS は NTFS、EXT3、ZFS 等、独自のディスク・ファイル・フォーマットを持つが、OVF レベルで記述または理解するにはこれらは重要ではない。

OVF 記述の例については、2.3 章および付録 A、B に記載する。

## 2.2 OVF 環境

デプロイした特定の環境で適切に機能させるために、仮想アプライアンスをカスタマイズする必要があることも多い。OVF 環境は、仮想化プラットフォームがゲスト・ソフトウェアにデプロイメント・コンフィギュレーションを伝えるために、標準的で拡張可能な方法を提供する。

OVF 環境は XML 文書であり、ゲスト・ソフトウェアのデプロイメント時に必要となるカスタマイズに関する情報を含む。本 XML 文書で提供できる情報の例を次に挙げる。

- ・ ホスト名、IP アドレス、サブネット、ゲートウェイ等のオペレーティング・システム・レベルでのコンフィギュレーション
- ・ アクティブ・ディレクトリ・サーバーの DNS 名、データベース、その他外部サービス等のアプリケーション・レベルのコンフィギュレーション

デプロイメントの際に設定すべきプロパティのセットは、**ProductSection** メタデータを使用して OVF 記述子に指定され、一般には、デプロイメントの際にユーザーがウィザード・スタイルのインターフェースを使用して入力する。

例えば、OVF 環境によって、ゲスト・ソフトウェアは多層サービス間のネットワーク設定を自動化することができ、ウェブ・サーバーは、ユーザーが手動で行わなくても、データベース・サーバーの IP アドレスを用いて自身の設定を自動的にできるようになる。

標準 OVF 環境を定義することは、いくつかの課題を提示することになる。なぜなら、仮想マシンで動作するゲスト・ソフトウェアと、基盤となる仮想化プラットフォームとの通信を目的とした、クロス・ベンダー型の準仮想化デバイスの標準は存在しないからである。OVF 仕様が取る方法は、OVF 環境の定義を次の 2 部に分けることである。i) どの情報が利用可能か、どのフォーマットでその情報が利用可能かを指定する標準プロトコルと、ii) どのように情報を入手するかを指定するトランスポートである。

本仕様は、すべての実装が ISO トランスポート（プロトコル）をサポートすることを要求しているが、その ISO トランスポートによって、動的に作成された ISO イメージ上でゲスト・ソフトウェアが OVF 環境（XML 文書で指定される）を利用できるようになる。

OVF 環境文書の例を、付録 A および B に記載する。

## 2.3 OVF 記述子の見本

次の記載は、一般的なシングル仮想マシン・アプライアンスの完全な OVF 記述子を示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://schemas.dmtf.org/ovf/1/envelope"
  xmlns:ovf="http://schemas.dmtf.org/ovf/1/envelope"
  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">

  <!-- References to all external files -->
  <References>
    <File ovf:id="file1" ovf:href="vmdisk1.vmdk" ovf:size="180114671"/>
  </References>

  <!-- Describes meta-information for all virtual disks in the package -->
  <DiskSection>
    <Info>Describes the set of virtual disks</Info>
    <Disk ovf:diskId="vmdisk1" ovf:fileRef="file1" ovf:capacity="4294967296"
  ovf:format="http://www.vmware.com/interfaces/specifications/vmdk.html#sparse"/>
  </DiskSection>

  <!-- Describes all networks used in the package -->
  <NetworkSection>
    <Info>List of logical networks used in the package</Info>
    <Network ovf:name="VM Network">
      <Description>The network that the service will be available on</Description>
    </Network>
  </NetworkSection>

  <VirtualSystem ovf:id="vm">
    <Info>Describes a virtual machine</Info>
```

```
<Name>Virtual Appliance One</Name>

<ProductSection>
  <Info>Describes product information for the appliance</Info>
  <Product>The Great Appliance</Product>
  <Vendor>Some Great Corporation</Vendor>
  <Version>13.00</Version>
  <FullVersion>13.00-b5</FullVersion>
  <ProductUrl>http://www.somegreatcorporation.com/greatappliance</ProductUrl>
  <VendorUrl>http://www.somegreatcorporation.com/</VendorUrl>
  <Property ovf:key="admin.email" ovf:type="string">
    <Description>Email address of administrator</Description>
  </Property>
  <Property ovf:key="app.ip" ovf:type="string" ovf:defaultValue="192.168.0.10">
    <Description>The IP address of this appliance</Description>
  </Property>
</ProductSection>

<AnnotationSection ovf:required="false">
  <Info>A random annotation on this service. It can be ignored</Info>
  <Annotation>Contact customer support if you have any problems</Annotation>
</AnnotationSection>

  <EulaSection>
    <Info>License information for the appliance</Info>
    <License>Insert your favorite license here</License>
  </EulaSection>

<VirtualHardwareSection>
  <Info>256MB, 1 CPU, 1 disk, 1 nic</Info>
  <Item>
    <rasd:Description>Number of virtual CPUs</rasd:Description>
    <rasd:ElementName>1 virtual CPU</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
  </Item>
  <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>
<Item>
  <rasd:AutomaticAllocation>true</rasd:AutomaticAllocation>
  <rasd:Connection>VM Network</rasd:Connection>
  <rasd:ElementName>Ethernet adapter on "VM Network"</rasd:ElementName>
  <rasd:InstanceID>4000</rasd:InstanceID>
  <rasd:ResourceType>10</rasd:ResourceType>
</Item>
<Item>
  <rasd:ElementName>Harddisk 1</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/vmdisk1</rasd:HostResource>
  <rasd:InstanceID>22001</rasd:InstanceID>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
</VirtualHardwareSection>
<OperatingSystemSection ovf:id="58" ovf:required="false">
  <Info>Guest Operating System</Info>
  <Description>Windows 2000 Advanced Server</Description>
</OperatingSystemSection>
</VirtualSystem>
</Envelope>
```

ほとんどの記述子は定型的な記述である。記述子自体に加えてファイル・セットの記述から始まる。本例では、1つのファイルしかない (vmdisk1.vmdk)。その次に、アプリケーションが使用する仮想ディスクのセットとネットワークのセットを記述する。各ファイル、ディスク、ネットワーク・リソースはユニークな識別子を与えられる。これらはすべて、個別の名前空間に置かれるが、最も良いのは識別しやすい名称を使うことである。



例として取り上げた OVF のコンテンツはシングル仮想マシンである。コンテンツには 5 つのセクションがある。

- アプライアンスの名称とベンダー、およびアプライアンスのカスタマイズに使用できるプロパティのセット、等の製品情報を提供する **ProductSection**。このプロパティはアプライアンスのインストール時に、通常はユーザーを介して設定される。この点に関しては別途詳細に説明する。
- 自由形式の注釈である **AnnotationSection**。
- アプライアンスのライセンス条項である **EULASection**。一般にインストール中に表示される。
- 仮想ハードウェアを記述する **HardwareSection**。このセクションは、仮想マシンが必要とする仮想ハードウェアとデバイス・セットの種類を記述するセクションである。この事例では、非常に一般的なハードウェアのセット (500MB のゲスト・メモリ、1CPU、1NIC および 1 つの仮想ディスク) が示されている。外部セクションからのネットワークとディスク識別子はここで参照される。
- ゲスト・オペレーティング・システムを記述する **OperatingSystemSection**。

### 3 オープン仮想化フォーマットの使用

#### 3.1 作成

OVF の作成には次の課程が含まれる。i) VM のセットを仮想ディスクのセットの上にパッケージ化する。ii) 前記の仮想ディスクを適切にエンコードする。iii) 仮想ハードウェア、ライセンシング、その他カスタマイズされたメタデータの記載を含む OVF 記述子を加える。iv) パッケージに任意でデジタル署名する。OVF のインストールまたはインポート・プロセスは、仮想化プラットフォームが OVF を取り込み、OVF のコンテンツから仮想マシンのセットを作成するときに生じる。

OVF は、既存の仮想マシンを仮想化プラットフォームから OVF パッケージにエクスポートし、正しくインストールして実行するために必要な、関連するメタデータを付与することで簡単に作成できる。このプロセスによって、仮想マシンを、特定のハイパーバイザー上の現在の実行状態から OVF パッケージに移行できる。このプロセスの間に、仮想マシンのディスクを配布しやすいように圧縮することができる。

商用品質の仮想アプライアンスについては、標準的に構築された環境を使用して OVF パッケージを作成してもよい。例えば、OVF 記述子はソース管理システムを使用することで管理でき、OVF パッケージは、(make ファイル等の) 再生可能なスクリプト環境、または複

数のベンダーから入手できるアプライアンス構築ツールキットを使用することで構築できる。

OVF を作成するときは、アプライアンス特有のインストール後のコンフィギュレーションメタデータが必要である。このメタデータには、アプライアンスのインターフェース言語への任意のローカライズ用、EULA（使用許諾ライセンス）のレビュー/サインオフおよび/または強化用、そしてリソース・コンフィギュレーション用のメタデータも含まれる。特別なドライバー、エージェント、および I/O、タイムキーピング、メモリ管理、監視、規則的なシャットダウンの強化等のゲストに対するその他のツールを含むこともある。

### 3.2 デプロイメント

デプロイメントによって、OVF パッケージ中の仮想マシンは、ターゲットとする仮想化プラットフォームが理解できる実行時フォーマットに変換され、適切なリソースを割り当てられ、正しい仮想ハードウェアによってサポートされる。デプロイメント中に、プラットフォームは OVF の整合性を検証して、OVF パッケージが送信中に改ざんされていないことを確認し、ローカル仮想ハードウェアと互換性があることを確認する。プラットフォームはまた、ターゲットとなる仮想化プラットフォーム固有の環境にあわせてリソースを割り当て、仮想マシンを設定する。この作業には仮想マシンが接続されなければならない（物理および仮想）ネットワークの割り当てと設定が含まれる。VM のストレージ・リソースを、仮想ハード・ディスクだけでなく、一時データセット、クラスター化またはネットワーク・ストレージへの接続等まで含めて割り当てし、CPU とメモリ・リソースを設定し、アプリケーション・レベルのプロパティをカスタマイズする。OVF はプロセッサ・アーキテクチャまたはハードウェア・プラットフォーム間のゲスト・ソフトウェアの変換をサポートしない。デプロイメントによって、OVF 記述子の要件に適合するハードウェア・プロファイルを持つ 1 つ以上の仮想マシンと、OVF パッケージが指定するコンテンツを持つ仮想ディスクのセットがインスタンス化される。

OVF パッケージのデプロイメント・エクスペリエンスは、それがデプロイされる仮想化プラットフォームによって異なる。コマンドライン・ベース、スクリプト形式、グラフィカル・デプロイメント・ウィザードを用いるもの等、様々である。OVF デプロイメント・ツールは一般に次の情報を表示するか、または入力プロンプトを表示する。

- OVF パッケージの（ProductSection からの）情報を示し、ユーザーにライセンス契約条項を承認するか、無人のインストールを処理するかを尋ねる。
- 仮想ハードウェアが OVF の仕様と互換性があるかを検証する。

- ユーザーに仮想マシンのストレージの場所を尋ね、OVF パッケージの論理ネットワークが接続すべき物理ネットワークを尋ねる。
- ユーザーに **ProductSection** で設定された特定のプロパティ値を入力するように伝える。

このコンフィギュレーションの後、仮想マシンは問題なく起動し、(DHCP 等の標準手続きを使用して) ローカル・ネットワークで有効な ID を入手できるはずである。プロパティを使用して、特定の IP ネットワーク・コンフィギュレーションや、その他のデプロイメント環境に特有な値を入力するプロンプトを表示する。一度アプライアンスをブートすれば、アプライアンスに格納されているソフトウェアを、ウェブ・インターフェース等のアプライアンス自体が提供する管理インターフェースを通じて、追加導入できる。

## 4 特徴

### 4.1 仮想ハードウェア記述

セクション 2.3 で示したハードウェア記述は非常に一般的なものである。具体的には、仮想ディスク 1 つとネットワークアダプタ 1 つが必要であることしか指定していない。特定のハードウェアが何であるべきかについても指定していない。例えば、ディスクが SCSI と IDE のどちらが、またはネットワークカードが E1000 と Vlanace のどちらが適切とされるべきか、は指定しない。さらに、仕様が一般的な場合は、アプライアンスが存在するデバイスを検知し、関連ドライバをロードすることが当然推定できる。この場合は、アプライアンスの作成者が広範なドライバー・セットを備えるアプライアンスを開発し、アプライアンスが動作することを確認するために関連仮想ハードウェア上でアプライアンスを試験済みである、としなければならない。

OVF パッケージを、同じハードウェア・デバイス、および/または、アプライアンスに含まれるゲスト OS が必要とする同じ種類のデバイスを提供しないプラットフォーム上にインストールする場合は、重大で原意不明のインストール・エラーが起こる可能性がある。アプライアンスが正しく動作しないというリスクだけではなく、インストールおよびブートに失敗し、ユーザーが問題を解決できないというリスクが生じる。このリスクに加えて、顧客からのサポートを求める電話が増加し、全体的な顧客の不満が高まるリスクも生じる。制約の多い、詳細な仮想ハードウェアの仕様を提供することによって、(必要となる特定のデバイスを記載するため) 不正確な実行を減らすことができるが、アプライアンスを正しくインストールできるシステムの数を制限することになる。

インストールの簡潔さ、堅牢性、予測可能性こそが、ISV が仮想アプライアンス・モデルに移行している主要な理由であることに留意すると、アプライアンス開発者は、アプライアンスが非常に特別なハードウェアを必要としない場合は、より一般的なハードウェア仕様を持つアプライアンスを開発するよう心がけるべきである。最初の段階では、アプライアンスのポータビリティは仮想マシンで使用されるゲスト OS に基づく。

アプライアンス・ベンダーは、自らが望むすべてのターゲット仮想化プラットフォームの、仮想ハードウェアに対するデバイス・ドライバーを持つ仮想マシンを開発することが理想的であろう。しかし、現在の仮想化プラットフォーム・ベンダーの多くは仮想アプライアンス・ベンダー/開発者にドライバーを個別に配布していない。その代わりに、仮想ハードウェア/アプライアンス・インターフェースの管理をさらに簡単にするために、OVF モデル

は明示的なインストール・モードをサポートする。このモードでは、各仮想マシンをインストール後すぐにブートでき、特定の仮想化プラットフォームに対するローカライズ/カスタマイズが許可される。これにより、仮想マシンは仮想化プラットフォームを検知し、正しいデバイス・ドライバーのセットをインストールすることができ、(例えば、初回ブート時にゲストに付属しているフロッピーまたは CD ドライブを使用して) 最初のリブート時にゲストが利用できるようになる、プラットフォーム固有ドライバーもインストールできる。さらに、sysprep 対応の Windows の VM は、再インストールおよびネーミング等のカスタマイズのみを必要とするが、リブート技術を使うと、イメージのネーミングとテーラリングが自動的に行われる。

複数の仮想ハードウェア・プロファイルが同じ記述子で指定されている例を示す。

```
<VirtualHardwareSection>
  <Info>500Mb, 1 CPU, 1 disk, 1 nic virtual machine</Info>
  <System>
    ...
  </System>
  <Item>
    ...
  </Item>
  ...
</VirtualHardwareSection>

<VirtualHardwareSection>
  <Info>500Mb, 1 CPU, 1 disk, 1 nic virtual machine</Info>
  <System>
    ...
  </System>
  <Item>
    ...
  </Item>
  ...
</VirtualHardwareSection>
```

前記の例では、様々な仮想化プラットフォームと特徴をサポートするように、ベンダーがバージョン 1.0.0

ハードウェア記述子を調整できる。特定の仮想化プラットフォームは、OVF インストーラがローカル・プラットフォーム上で利用可能な最新、または最良の機能を選ぶという前提に基づき、その仮想化プラットフォームがサポートできる仮想ハードウェア・セクションのいずれかを選ぶことができる。

特定のデバイス・タイプが指定されている例を示す。

```
<Item>
  <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
  <rasd:InstanceID>1000</rasd:InstanceID>
  <rasd:ResourceSubType>LsiLogic BusLogic</rasd:ResourceSubType>
  <rasd:ResourceType>6</rasd:ResourceType>
</Item>
<Item>
  <rasd:ElementName>Harddisk 1</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/vmdisk1</rasd:HostResource>
  <rasd:InstanceID>22001</rasd:InstanceID>
  <rasd:Parent>1000</rasd:Parent>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
```

前記の例では、ResourceSubType を使用して、アプライアンスのゲスト OS がサポートする正確なデバイスを指定する。

## 4.2 デプロイメント・オプション

OVF パッケージの作者は、仮想アプライアンスの対象となるリソース要件に関するメタデータを含むことができる。これは次のように、人間が読める形でのコンフィギュレーション・リストとしてフォーマットされる。

1. ソフトウェア評価設定
2. 10 人から 100 人までのワークグループの設定
3. 100 人から 1000 人までのワークグループの設定
4. 大企業用ワークグループの設定

パッケージのデプロイメント担当者は、デプロイメント中にコンフィギュレーションを選

ぶようプロンプト表示される。正確な値に加えて、値の範囲も指定することができる。例えば、メモリサイズを **600MB** に指定し、推奨範囲が **500MB** から **1000MB** であるように指定することもできる。一般には、ユーザーは **OVF** パッケージの導入中に、範囲の値を指定するようにプロンプト表示されることはない。前記のコンフィギュレーション・リストは、初期リソース・コンフィギュレーションを良くするするために使用することが想定されている。範囲の指定は、異なるリソースのニーズに基づき、後ほどインストールの変更が必要となった場合に役立つ。

コンフィギュレーション・リストの例。

```
<DeploymentOptionSection>
  <Configuration ovf:id="min">
    <Label>Minimal</Label>
    <Description>Minimal setup</Description>
  </Configuration>
  <Configuration ovf:id="normal" ovf:default="yes">
    <Label>Normal</Label>
    <Description>Standard setup</Description>
  </Configuration>
  ... more configurations ...
</DeploymentOptionSection>
```

リソース要件の例。

```
<ResourceAllocationSection>
  <Info>Defines reservations for CPU and memory</Info>
  <Item>
    ... normal configuration ...
  </Item>
  <Item ovf:configuration="min">
    ... overwrites for minimal configuration ...
  </Item>
</ResourceAllocationSection>
```

VirtualHardwareSection の例。

```
<VirtualHardwareSection>
  <Info>...</Info>
  <Item>
    <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
    <rasd:ElementName>1 CPU and 500 MHz reservation</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:Reservation>500</rasd:Reservation>
    <rasd:ResourceType>4</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
  </Item>
  ...
  <Item ovf:configuration="big">
    <rasd:ElementName>1 CPU and 800 MHz reservation</rasd:ElementName>
    <rasd:InstanceID>0</rasd:InstanceID>
    <rasd:Reservation>600</rasd:Reservation>
    <rasd:ResourceType>3</rasd:ResourceType>
  </Item>
</VirtualHardwareSection>
```

### 4.3 デプロイメントのカスタマイズ

OVF 記述子は、ゲストにインストールされたソフトウェア製品の記述を、OVF 環境を通じてどのようにカスタマイズされるかもあわせて、含むことができる。

```
<ProductSection>
  <Info>Describes product information for the service</Info>
  <Product>MyService Web Portal</Product>
  <Vendor>Some Random Organization</Vendor>
  <Version>4.5</Version>
  <FullVersion>4.5-b4523</FullVersion>
  <ProductUrl>http://www.vmware.com/go/ovf</ProductUrl>
  <VendorUrl>http://www.vmware.com/</VendorUrl>
  <Property ovf:key="adminEmail" ovf:type="string" ovf:userConfigurable="true">
    <Description>Email address of administrator</Description>
  </Property>
```



```
<Property ovf:key="appIp" ovf:type="string" ovf:userConfigurable="true">
  <Description>IP address of the application</Description>
</Property>
</ProductSection>
```

**Property** エlementはアプリケーション・レベルのカスタマイズ・パラメータを特定し、ネットワーク ID、DNS サーバーやゲートウェイの IP アドレス、その他の設定などを、デプロイメント中にカスタマイズする必要があるアプライアンスに特に関連する。

付録 A は複合型多層アプリケーションの詳細なカスタマイズ例を記載する。

#### 4.4 国際化

OVF 仕様はオプションの `ovf:msgid` 属性を使用してローカライズ可能なメッセージをサポートする。

```
<Envelope ...>
  ...
  <Info ovf: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>
</Envelope>
```

前記の例では、ローカライズされた文字列は **OVF** 記述子の中に保存されているが、ローカライズされた文字列は **OVF** 記述子の外に外部文字列バンドルを使用して保存されることもある。例を次に挙げる。

```
<Envelope ...>
```

```

<References>
  ...
  <File ovf:id="da-DA-resources" ovf:href="german.msg"/>
  <File ovf:id="de-DE-resources" ovf:href="danish.msg"/>
  ...
</References>
  ...
  <Strings xml:lang="da-DA" ovf:fileRef="da-da-resources"/>
  <Strings xml:lang="de-DE" ovf:fileRef="de-de-resources"/>
</Envelope>

```

#### 4.5 拡張可能性

OVF 仕様の設計目標は、下位および上位互換性を確保することである。下位互換性とは、i) 仕様の以前のバージョンで書かれた、または ii) 特定の拡張子の知識を持たない OVF 利用者が、将来の仕様の特徴を利用した OVF 記述子（またはカスタム拡張子）を理解できることを意味している。OVF 利用者は信頼性、予測性が高く、使いやすい方法で、拡張子を含む OVF パッケージを拒絶するか承認するかを決定できるべきである。

OVF は、既存のセクションが新しいコンテンツを含んで拡張することを許可するだけでなく、セクションを追加することも許可するオープン・コンテンツ・モデルをサポートする。拡張に際しては、ブーリアン属性である `ovf:required` が、エレメントの情報が正しい動作に必須か、または任意であるかを指定する。

新セクションを追加する例：

```

<ns:BuildInformationSection ovf:required="false">
  <Info>Specifies information on how a virtual machine was created</Info>
  <BuildNumber> ... </BuildNumber >
  <BuildDate> ... </BuildDate >
  <BuildSystem> ... </BuildSystem>
  ...
</ns:BuildInformationSection>

```

既存のセクションを拡張する例：

```

<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 -->

<ns:Author ovf:required="false">John Smith</ns:Author>

</AnnotationSection>
```

OVF 文書の拡張子に関する詳細な例は付録 C に記載する。

#### 4.6 適合性

OVF 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 OVF 仕様で定義するメタデータのみを含む。つまりカスタム拡張子は存在しない。

適合性レベル : 1

- OVF 記述子はカスタム拡張子を持つメタデータを含むが、その拡張子はすべて任意である。

適合性レベル : 2

- OVF 記述子はカスタム拡張子を持つメタデータを含むが、少なくともその 1 つの拡張子は必須である。

適合性レベル : 3

適合性レベル 3 の使用はポータビリティを制限するため、可能な限り避けるべきである。

### 5 ポータビリティ

OVF は仮想アプライアンスと関連する仮想マシンのポータビリティを高めることができる技術である。OVF パッケージには、仮想化プラットフォームが簡潔に解釈できる仮想マシンを生成する方法が含まれる。パッケージ化したメタデータによって、仮想マシンを確実に使いやすい方法でインストールできるようになる。具体的には、メタデータを使用することで、管理インフラストラクチャーは、OVF で記述する特定の VM をインストールできるか、または拒絶すべきかを確実に決定することができ、VM がインストールされる特定の実行コンテキストにおいて動作できるようにするために、適切な変換とローカライズを選ぶことができる。

OVF フォーマット仕様では管理できない多くの要因もあり、OVF フォーマット仕様に完全バージョン 1.0.0

に準拠して実装したとしても、パッケージ化された仮想マシンのポータビリティを決定する多くの要因がある。つまり、仮想マシンを OVF パッケージにパッケージ化することで、すべてのハイパーバイザーでインストールできる普遍的なポータビリティは保証されないということである。次に挙げる要因はポータビリティを制限する可能性のある例である。

- ・ OVF 内の VM が、インストールを試みるハイパーバイザーが理解できないフォーマットの仮想ディスクを含むことがある。ほとんどのハイパーバイザーが、主要な仮想ハードディスク・フォーマットのいずれかを用いる VM をインポートおよび/またはエクスポートできると期待することは当然であるが、特定のハイパーバイザーでサポートされず、OVF がサポートする新しいフォーマットが生じることもある。本仕様の将来のバージョンで、OVF 準拠のハイパーバイザーがサポートすべきハードディスク・フォーマットのセットを規定することも有効である。

- ・ インストールされたゲスト・ソフトウェアが、ハイパーバイザーが提示する仮想ハードウェアをサポートしないこともある。例えば、Xen ハイパーバイザーは仮想フロッピーディスク・デバイスをゲストに提供するにはデフォルト設定されていない。フロッピーディスク・コントローラとのやりとりを要求するゲスト VM があると、VM を正しく実行できないこともある。

- ・ インストールされたゲスト・ソフトウェアが CPU アーキテクチャをサポートしない。例えば、ゲスト・ソフトウェアがあるプロセッサ・モデルに特有の CPU 操作を実行し、または特定の浮動小数点サポートを必要とし、または特定のベンダー CPU に特定の命令コードを含むこともある。

- ・ 仮想化プラットフォームが、OVF 記述子が要求する特徴を理解しないこともある。例えば、複合型サービスはサポートされないこともある。OVF 標準は仮想化製品とは独立して進展するため、いずれかの時点で、OVF 仕様以前の仮想化プラットフォーム上でその OVF がサポートされないこともある。

OVF のポータビリティは次の 3 レベルに分けることができる。

- ・ レベル 1。特定の仮想化製品および/または CPU アーキテクチャおよび/または仮想ハードウェア上でのみ動作する。これは一般的に、OVF が停止中の仮想マシン、または電源がオンの仮想マシンのスナップショットを含み、CPU、および実デバイスまたはエミュレート・デバイスなどの現在の実行時状態を含むことに起因する。前記の状態では、OVF は非常に特化された仮想化およびハードウェア・プラットフォームに結びつけられる。

- ・ レベル 2。仮想ハードウェアの特定のファミリー上で動作する。これは、インストール

されたゲスト・ソフトウェアによるドライバーのサポートがないことが原因であることが多い。

・ レベル 3。仮想ハードウェアの複数のファミリー上で動作する。例えば、アプライアンスが Xen、Sun、Microsoft、VMware のハイパーバイザー上で動作可能な場合である。レベル 3 の互換性を持つために、ゲスト・ソフトウェアは複数のハイパーバイザーのデバイスをサポートするように開発されている。ゲスト OS を問題なくインストールしてブートする間に、ゲスト OS がハードウェア・デバイスの探知を行い、仮想化プラットフォームとやりとりするのに必要な特定のドライバーをインストールすることは、OVF のポータビリティレベル 3 の例である。Microsoft Windows® オペレーティング・システム用の sysprep 対応のポータビリティのレベルはもうひとつの例である。このような OS のインスタンスは、仮想ハードウェアを含む複数のハードウェア・プラットフォーム上で再インストール、再ネーミング、再パーソナライズできる。

組織内で使用するには、OVF パッケージが統制された環境内で配布され、特定のハードウェアや仮想化プラットフォームを購入する決定によって、OVF の基礎となる特徴の一貫性を保証できるため、レベル 1 またはレベル 2 の互換性で十分である。仮想マシンを単にエクスポートすることは一般的にはレベル 1 またはレベル 2 の互換性を持つ、(特定の仮想ハードウェア・セットに結びつけられた) OVF を作成するが、レベル 3 互換のエクスポート機能をサポートするようにメタファーを拡張することは、例えば Windows の「sysprep」等のツールを用いれば簡単にできる。

ISV が独自に作成し配布する商用アプライアンスについては、レベル 3 の互換性が強く望まれる。実際に、レベル 3 の互換性によって、評価および製造のために、アプライアンスをすぐに利用できる最大限の顧客ベースが確保できる。認証された「検証済み」レベル 3 のアプライアンス・パッケージを作成し、複数の仮想化プラットフォームに広範囲に配布し、インストールするためのツールキットが、あるいはアプライアンスが、より範囲の狭い仮想ハードウェア・セットで使用される場合、例えば企業内の特定の開発グループの中で使用される場合は、レベル 2 互換のパッケージを作成するツールキットが一般に使用されるだろう。

OVF 仮想ハードウェア記述はレベル 1 からレベル 3 までのポータビリティをサポートするように設計されている。レベル 3 のポータビリティでは、ハードウェア要件は非常に一般的な記述しか含めないか、または複数の代替仮想ハードウェア記述を指定することができる。アプライアンスのプロバイダーは仮想ハードウェアの仕様を完全に管理し、柔軟にす

ることも、制約を多くすることもできる。範囲の狭い仕様を用いてアプライアンスを制約し、動作検証済みの仮想ハードウェア上でのみ動作させることもできるが、そのポータビリティをある程度制限することになる。広範囲の仕様では、可能な限り幅広い仮想ハードウェア上でアプライアンスが使用できる。これによって、顧客は最高のユーザー・エクスペリエンスを持つことができるが、これが仮想アプライアンス概念を成功させる主要な要件の1つである。

## 6 OVF 仕様の将来のバージョン

OVF 仕様バージョン 1.0 の範囲は仮想アプライアンス・ソフトウェア・ライフサイクルのパッケージ化段階とデプロイメント段階である。OVF1.0 は、ワークフローとシステム・レベルのメタデータをエンコードし、保存し、トランスポートするための中心となる枠組みを提供することである。

OVF パッケージには、アプライアンスが外部のプロセスおよびシステムとの対話方法を記述する情報を保存することができる。この機能の例は、アプライアンスの更新、カタログ化、整合性および/またはセキュリティのチェック、依存度チェック、および強化したライセンス管理等である。仕様の将来のバージョンでは前記のメタデータの標準化を取り扱う予定である。

OVF パッケージは、複合ネスト化コンフィギュレーションを含む多層アプリケーションを持つことができるが、OVF は現在のところ既存の OVF パッケージの組み込みをサポートしていない。既存の署名済み OVF パッケージのソフトウェアを新しいコンテキストに埋め込むときには、既存のパッケージを組み込むことは魅力的である。仕様の将来のバージョンではこの点のサポートをとりあげる予定である。

## 7 結論

OVF 仕様は IT 業界すべてで幅広く導入されることを目的とした、ポータビリティのある仮想アプライアンス・フォーマットを提供する。OVF 仕様はすぐに役立ち、緊急のビジネス・ニーズを解決することを目的とし、また、一般的で後方互換性を持つ、優れた仮想マシン・フォーマットの迅速な採用を促進することを目的としている。OVF は既存の IT の管理標準やフレームワークと相互に補完し、標準化団体でさらに開発が進められている。OVF は様々なベンダーの仮想化プラットフォーム間での仮想マシンのポータビリティと交換の共通な標準を協力して開発することによって顧客の信頼を高め、オープン性と拡張性により、最高の製品の競争を促進する。

OVF 仕様は適切な標準化団体で発展することを意図している。本文書に添付された明確な著作権の表明は、標準化団体に関知しない、本フォーマットに対する独自拡張を避けるためであるが、仕様の配布と実装は自由である。

## A 多層 Petstore 例

本例では最新の OVF 概念をいくつか紹介する。

複数 VM パッケージ—VirtualMachineCollection エンティティ・サブタイプを使用する。  
複合サービス団体—ネスト化した VirtualMachineCollection エンティティ・サブタイプを使用する。

- ・ ユーザー定義のデプロイメントコンフィギュレーションを伝播する。
- ・ OVF 環境を使用するサービスをデプロイメント時にカスタマイズする。
- ・ ダウンロードを最低限に抑えるために仮想ディスク・チェーンを使用する。

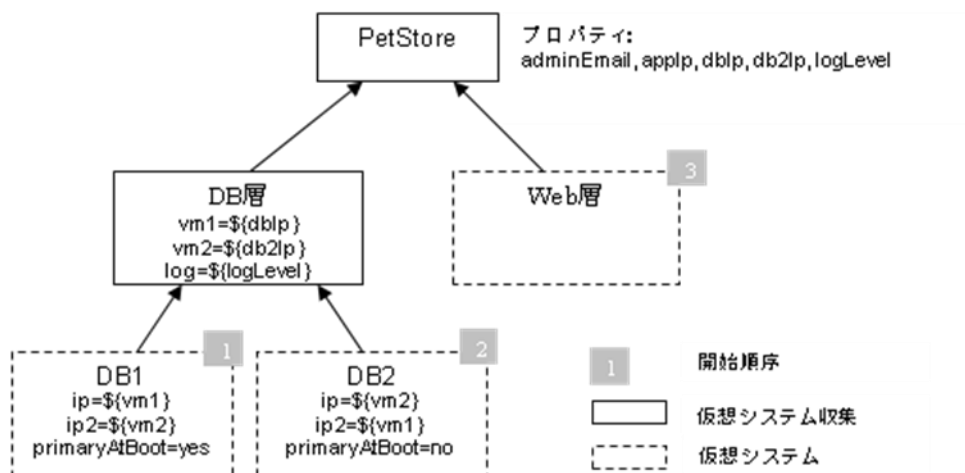
個々の仮想マシンにインストールしたソフトウェアに関する情報を提供する ProductSection をネスト化する。

サービスの例は PetStore と呼ばれ、フロントエンド・ウェブサーバーとデータベースから成る。データベース・サーバーはそれ自体複合型多層サーバーであり、耐障害性のために 2 つの VM から成る。

### アーキテクチャーとパッケージ化

Petstore OVF パッケージは 3 つの仮想システム (Web 層、DB1、DB2) と 2 つの仮想システム・コレクション (Petstore と DB 層) から成る。次の図は OVF パッケージの構造と、仮想マシンのプロパティと開始順序を示す。





完全な OVF 記述子は本文書の後半に記載する。OVF のプロパティとディスク・レイアウトの使用は次に詳細に検討する。

### プロパティ

Petstore サービスは 5 つのユーザーが設定可能なプロパティを持つ。これらはデプロイされた環境で正しく開始するために、設定する必要があるサービスの主要な管理パラメータである。プロパティは OVF 環境文書の形でゲスト・ソフトウェアまで伝達される。ゲスト・ソフトウェアは起動時に OVF 環境を読み取り、プロパティの値を抽出し、そしてそれらをソフトウェアのコンフィギュレーションに適用するように書かれている。従って、OVF 記述子はゲスト・ソフトウェアが処理するプロパティを反映する。

この特定のサービスのために、Web 層とデータベース層向けの 2 種類のソフトウェア・コンフィギュレーションがある。それぞれのソフトウェア・コンフィギュレーションでサポートするプロパティは次の通りである。

### ウェブゲスト・ソフトウェア

プロパティ	記述
appIp	ウェブ・サーバーの IP アドレス
dbIp	接続するデータベース・サーバーの IP アドレス

	ス
adminEmail	サポートする E メール・アドレス
logLevel	ロギング・レベル

直接の親 `VirtualSystemCollection` コンテナ上で定義されるすべてのプロパティは子 `VirtualSystem` または `VirtualSystemCollection` でも利用可能である。従って、OVF 記述子は Web VM 用に提示されるように、各 VM の明白な `ProductSection` を含む必要はない。

### データベースゲスト・ソフトウェア

プロパティ	記述
Ip	仮想マシンの IP アドレス
primaryAtBoot	ブート時にインスタンスがプライマリとして動作するかセカンダリとして動作するか。
ip2	ホット・スペアまたはプライマリとして動作するツイン・データベース VM の IP アドレス。
log	ここでロギング・レベルは <code>log</code> と呼ばれる。

クラスター化されたデータベースは仮想システム・コレクション自体として編成され、コンフィギュレーション用の特定のプロパティセット、`vm1`、`vm2`、および `log` を持つ。この編成によって、データベースの実装が OVF パッケージの他のソフトウェアから区別され、仮想アプライアンス（ゲスト・ソフトウェアと仮想マシン・コンフィギュレーション）が簡単に構成でき、それに伴って再利用を促進する。

データベースソフトウェアは市販のソフトウェア・パッケージであり、ベンダーは「`com.mydb.db`」をすべてのプロパティのユニークネームとして選択してきた。これは、OVF 記述子で `ProductSection` の `ovf:class` 属性を含めて、見ることができる。

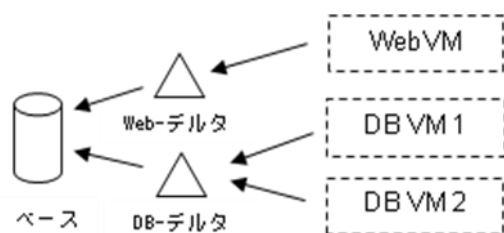
`#{<name>}` プロパティ・シンタックスは、外部レベルの値を OVF 記述子のエンティティ階層の内部ノードに伝播するために使用される。このメカニズムを使うことで、ネーミング転換やゲスト・ソフトウェアの変更を事前調整することなく、様々な構成要素をリンクできる。直接の親 `VirtualSystemCollection` コンテナ上で定義されたプロパティのみが、子エンティティでも利用できる。従って、Petstore 上で定義されたプロパティは DB1 では利用できない。このため、`VirtualSystemCollection` のインターフェースがその親である `VirtualSystemCollection` に内包され、記述されていることが保証されることで、ソフトウェアバージョン 1.0.0

エアを構成することができ、再利用がしやすくなる。

OVF 記述子は固定した非ユーザー割り当てプロパティを使用して、二つのデータベース仮想マシンが最初は同一のソフトウェア・イメージをブートしていたとしても、異なるロールにブートするようにする。`com.mydb.db.primaryAtBoot` という名称のプロパティは固定、非ユーザー設定可能な値で特定されるが、2つのイメージの値は異なる。ソフトウェアはこれをブート時に検知し、それによって操作をカスタマイズする。

### ディスクのレイアウト

Petstore OVF パッケージは、パッケージのサイズを最小化し、その結果パッケージのダウンロード時間も最小化するために、ディスクを共有し、デルタ・ディスク階層をエンコードする能力を使用する。この事例では、2つの異なるイメージしか持たず（データベースとウェブ）、どちらのイメージも同じベース OS 配布上に構築されているとすると、これらのイメージは OVF 記述子に次のようにエンコードできる。



従って、パッケージは3つの明白な仮想マシンを含むが、ダウンロード・サイズの合計は大幅に小さくなる。実際に、1つの完全な VM と、2つの比較的小さなデルタをダウンロードする必要がある。

デプロイメントシステム上の仮想ディスクを物理的にレイアウトすることは、OVF パッケージのディスク構造に依存しない。OVF パッケージは仮想ディスクとコンテンツ・サイズ（つまり、ディスク上で必要なビット）を記述する。OVF パッケージはまた、各仮想マシンが独立したディスクを持たなければならないことも指定する。従って、各仮想マシンが初期ブートで記述されたコンテンツを持つディスクを見る限り、また1つの仮想マシンが書き込んだ変更が他の仮想マシンに影響しない限り、仮想化プラットフォームは前記パッケージを3つの独立したフラットディスクを持つ3つの VM としてインストールでき、または、前記の編成、あるいは何か別の編成を複製することを選択できる。

## 完全な OVF 記述子

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  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"
  <!-- References to all external files -->
  <References>
    <File ovf:id="base" ovf:href="base.vmdk" ovf:size="180114671"/>
    <File ovf:id="webdelta" ovf:href="webapp-delta.vmdk" ovf:size="123413"/>
    <File ovf:id="dbdelta" ovf:href="dbapp-delta.vmdk" ovf:size="343243"/>
  </References>
  <!-- Describes meta-information about all virtual disks in the package.
  This example is encoded as a delta-disk hierarchy.
  -->
  <DiskSection>
    <Info>Describes the set of virtual disks</Info>
    <Disk ovf:diskId="base" ovf:fileRef="base" ovf:capacity="4294967296"
      ovf:populatedSize="1924967692"
      ovf:format="http://www.vmware.com/specifications/vmdk.html#streamOptimized"/>
    <Disk ovf:diskId="web" ovf:fileRef="webappdelta" ovf:parentRef="base"
      ovf:capacity="4294967296"
      ovf:format="http://www.vmware.com/specifications/vmdk.html#streamOptimized"/>
    <Disk ovf:diskId="db" ovf:fileRef="dbdelta" ovf:parentRef="base"
      ovf:capacity="4294967296"
      ovf:format="http://www.vmware.com/specifications/vmdk.html#streamOptimized"/>
  </DiskSection>
  <!-- Describes all networks used in the package -->
```

```
<NetworkSection>
  <Info>List of logical networks used in the package</Info>
  <Network ovf:name="VM Network">
    <Description ovf:msgid="network.description">The network that the service
      will be available on</Description>
  </Network>
</NetworkSection>
<!-- Deployment options for the packages -->
<DeploymentOptionSection>
  <Info>List of deployment options available in the package</Info>
  <Configuration ovf:id="minimal">
    <Label ovf:msgid="minimal.label">Minimal</Label>
    <Description ovf:msgid="minimal.description">Deploy service with minimal
      resource use</Description>
  </Configuration>
  <Configuration ovf:id="standard" ovf:default="true">
    <Label ovf:msgid="standard.label">Standard</Label>
    <Description ovf:msgid="standard.description">Deploy service with standard
      resource use</Description>
  </Configuration>
</DeploymentOptionSection>
<!-- PetStore Virtual System Collection -->
<VirtualSystemCollection ovf:id="PetStore">
  <Info>The packaging of the PetStoreService multi-tier application</Info>
  <Name>PetStore Service</Name>
  <!-- Overall information about the product -->
  <ProductSection>
    <Info>Describes product information for the service</Info>
    <Product>PetStore Web Portal</Product>
    <Vendor>Some Random Organization</Vendor>
    <Version>4.5</Version>
    <FullVersion>4.5-b4523</FullVersion>
    <ProductUrl>http://www.vmware.com/go/ovf</ProductUrl>
    <VendorUrl>http://www.vmware.com/</VendorUrl>
```

```
<Category ovf:msgid="category.email">Email properties</Category>
<Property ovf:key="adminEmail" ovf:type="string" ovf:userConfigurable="true">
  <Label ovf:msgid="property.email.label">Admin email</Label>
  <Description ovf:msgid="property.email.description">Email address of
    service administrator</Description>
</Property>
<Category ovf:msgid="category.network">Network properties</Category>
<Property ovf:key="appIp" ovf:type="string"
  ovf:userConfigurable="true">
  <Label ovf:msgid="property.appip.label">IP</Label>
  <Description ovf:msgid="property.appip.description">IP address of the
    service</Description>
</Property>
<Property ovf:key="dbIp" ovf:type="string" ovf:userConfigurable="true">
  <Label ovf:msgid="property.dpip.label">IP for DB</Label>
  <Description ovf:msgid="property.dpip.description">Primary IP address of
    the database</Description>
</Property>
<Property ovf:key="db2Ip" ovf:type="string"
  ovf:userConfigurable="true">
  <Label ovf:msgid="property.dpip2.label">IP for DB2</Label>
  <Description ovf:msgid="property.dpip2.description">A secondary IP
    address for the database</Description>
</Property>
<Category ovf:msgid="category.logging">Logging properties</Category>
<Property ovf:key="logLevel" ovf:type="string" ovf:value="normal"
  ovf:userConfigurable="true">
  <Label ovf:msgid="property.loglevel.label">Loglevel</Label>
  <Description ovf:msgid="property.loglevel.description">Logging level for
    the service</Description>
  <Value ovf:value="low" ovf:configuration="minimal"/>
</Property>
</ProductSection>
<AnnotationSection ovf:required="false">
```

```
<Info>A annotation on this service</Info>

<Annotation ovf:msgid="annotation.annotation">Contact customer support for
  any urgent issues</Annotation>

</AnnotationSection>

<ResourceAllocationSection ovf:required="false">

  <Info>Defines minimum reservations for CPU and memory</Info>

  <Item>

    <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>

    <rasd:ElementName>512 MB reservation</rasd:ElementName>

    <rasd:InstanceID>0</rasd:InstanceID>

    <rasd:Reservation>512</rasd:Reservation>

    <rasd:ResourceType>4</rasd:ResourceType>

  </Item>

  <Item ovf:configuration="minimal">

    <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>

    <rasd:ElementName>384 MB reservation</rasd:ElementName>

    <rasd:InstanceID>0</rasd:InstanceID>

    <rasd:Reservation>384</rasd:Reservation>

    <rasd:ResourceType>4</rasd:ResourceType>

  </Item>

  <Item>

    <rasd:AllocationUnits>MHz</rasd:AllocationUnits>

    <rasd:ElementName>1000 MHz reservation</rasd:ElementName>

    <rasd:InstanceID>1</rasd:InstanceID>

    <rasd:Reservation>500</rasd:Reservation>

    <rasd:ResourceType>3</rasd:ResourceType>

  </Item>

  <Item ovf:bound="min">

    <rasd:AllocationUnits>MHz</rasd:AllocationUnits>

    <rasd:ElementName>500 MHz reservation</rasd:ElementName>

    <rasd:InstanceID>1</rasd:InstanceID>

    <rasd:Reservation>500</rasd:Reservation>

    <rasd:ResourceType>3</rasd:ResourceType>

  </Item>

</ResourceAllocationSection>
```

```
<Item ovf:bound="max">
  <rasd:AllocationUnits>MHz</rasd:AllocationUnits>
  <rasd:ElementName>1500 MHz reservation</rasd:ElementName>
  <rasd:InstanceID>1</rasd:InstanceID>
  <rasd:Reservation>1500</rasd:Reservation>
  <rasd:ResourceType>3</rasd:ResourceType>
</Item>
</ResourceAllocationSection>
<StartupSection>
  <Info>Specifies how the composite service is powered-on and off</Info>
  <Item ovf:id="DBTier" ovf:order="1" ovf:startDelay="120"
    ovf:startAction="powerOn" ovf:waitingForGuest="true" ovf:stopDelay="120"
    ovf:stopAction="guestShutdown"/>
  <Item ovf:id="WebTier" ovf:order="2" ovf:startDelay="120"
    ovf:startAction="powerOn" ovf:waitingForGuest="true" ovf:stopDelay="120"
    ovf:stopAction="guestShutdown"/>
</StartupSection>
<VirtualSystem ovf:id="WebTier">
  <Info>The virtual machine containing the WebServer application</Info>
  <ProductSection>
    <Info>Describes the product information</Info>
    <Product>Apache Webserver</Product>
    <Vendor>Apache Software Foundation</Vendor>
    <Version>6.5</Version>
    <FullVersion>6.5-b2432</FullVersion>
  </ProductSection>
  <OperatingSystemSection ovf:id="97">
    <Info>Guest Operating System</Info>
    <Description>Linux 2.4.x</Description>
  </OperatingSystemSection>
  <VirtualHardwareSection>
    <Info>256 MB, 1 CPU, 1 disk, 1 nic virtual machine</Info>
    <System>
      <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
```



```
<vssd:InstanceID>0</vssd:InstanceID>
<vssd:VirtualSystemType>vmx-04</vssd:VirtualSystemType>
</System>
<Item>
  <rasd:Description>Number of virtual CPUs</rasd:Description>
  <rasd:ElementName>1 virtual CPU</rasd:ElementName>
  <rasd:InstanceID>1</rasd:InstanceID>
  <rasd:ResourceType>3</rasd:ResourceType>
  <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
</Item>
<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>
<Item>
  <rasd:AutomaticAllocation>true</rasd:AutomaticAllocation>
  <rasd:Connection>VM Network</rasd:Connection>
  <rasd:ElementName>Ethernet adapter on "VM Network"</rasd:ElementName>
  <rasd:InstanceID>3</rasd:InstanceID>
  <rasd:ResourceSubType>PCNet32</rasd:ResourceSubType>
  <rasd:ResourceType>10</rasd:ResourceType>
</Item>
<Item>
  <rasd:AddressOnParent>1</rasd:AddressOnParent>
  <rasd:ElementName>SCSI Controller 0 - LSI Logic</rasd:ElementName>
  <rasd:InstanceID>1000</rasd:InstanceID>
  <rasd:ResourceSubType>LsiLogic</rasd:ResourceSubType>
  <rasd:ResourceType>6</rasd:ResourceType>
</Item>
<Item>
```

```
<rasd:AddressOnParent>0</rasd:AddressOnParent>

<rasd:ElementName>Harddisk 1</rasd:ElementName>

<rasd:HostResource>ovf:/disk/web</rasd:HostResource>

<rasd:InstanceID>22001</rasd:InstanceID>

<rasd:Parent>1000</rasd:Parent>

<rasd:ResourceType>17</rasd:ResourceType>

</Item>

</VirtualHardwareSection>

</VirtualSystem>

<!-- Database Tier -->

<VirtualSystemCollection ovf:id="DBTier">

  <Info>Describes a clustered database instance</Info>

  <ProductSection ovf:class="com.mydb.db">

    <Info>Product Information</Info>

    <Product>Somebody Clustered SQL Server</Product>

    <Vendor>TBD</Vendor>

    <Version>2.5</Version>

    <FullVersion>2.5-b1234</FullVersion>

    <Property ovf:key="vm1" ovf:value="{dbIp}" ovf:type="string"/>

    <Property ovf:key="vm2" ovf:value="{db2Ip} " ovf:type="string"/>

    <Property ovf:key="log" ovf:value="{logLevel}" ovf:type="string"/>

  </ProductSection>

  <StartupSection>

    <Info>Specifies how the composite service is powered-on and off</Info>

    <Item ovf:id="DB1" ovf:order="1" ovf:startDelay="120"

      ovf:startAction="powerOn" ovf:waitingForGuest="true"

      ovf:stopDelay="120" ovf:stopAction="guestShutdown"/>

    <Item ovf:id="DB2" ovf:order="2" ovf:startDelay="120"

      ovf:startAction="powerOn" ovf:waitingForGuest="true"

      ovf:stopDelay="120" ovf:stopAction="guestShutdown"/>

  </StartupSection>

  <!-- DB VM 1 -->

  <VirtualSystem ovf:id="DB1">

    <Info>Describes a virtual machine with the database image installed</Info>
```

```
<Name>Database Instance I</Name>
<ProductSection ovf:class="com.mydb.db">
  <Info>Specifies the OVF properties available in the OVF environment</Info>
  <Property ovf:key="ip" ovf:value="{vm1}" ovf:type="string"/>
  <Property ovf:key="ip2" ovf:value="{vm2} " ovf:type="string"/>
  <Property ovf:key="primaryAtBoot" ovf:value="yes" ovf:type="string"/>
</ProductSection>
<VirtualHardwareSection>
  <Info>256 MB, 1 CPU, 1 disk, 1 nic virtual machine</Info>
  <System>
    <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
    <vssd:InstanceID>0</vssd:InstanceID>
    <vssd:VirtualSystemType>vmx-04</vssd:VirtualSystemType>
  </System>
  <Item>
    <rasd:Description>Number of virtual CPUs</rasd:Description>
    <rasd:ElementName>1 virtual CPU</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
  </Item>
  <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>
  <Item>
    <rasd:AutomaticAllocation>true</rasd:AutomaticAllocation>
    <rasd:Connection>VM Network</rasd:Connection>
    <rasd:ElementName>Ethernet adapter on "VM Network"</rasd:ElementName>
    <rasd:InstanceID>3</rasd:InstanceID>
```

```
<rasd:ResourceSubType>PCNet32</rasd:ResourceSubType>
<rasd:ResourceType>10</rasd:ResourceType>
</Item>
<Item>
  <rasd:AddressOnParent>1</rasd:AddressOnParent>
  <rasd:ElementName>SCSI Controller 0 - LSI Logic</rasd:ElementName>
  <rasd:InstanceID>1000</rasd:InstanceID>
  <rasd:ResourceSubType>LsiLogic</rasd:ResourceSubType>
  <rasd:ResourceType>6</rasd:ResourceType>
</Item>
<Item>
  <rasd:AddressOnParent>0</rasd:AddressOnParent>
  <rasd:ElementName>Harddisk 1</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/db</rasd:HostResource>
  <rasd:InstanceID>22001</rasd:InstanceID>
  <rasd:Parent>1000</rasd:Parent>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
</VirtualHardwareSection>
<OperatingSystemSection ovf:id="97">
  <Info>Guest Operating System</Info>
  <Description>Linux 2.4.x</Description>
</OperatingSystemSection>
</VirtualSystem>
<!-- DB VM 2 -->
<VirtualSystem ovf:id="DB2">
  <Info>Describes a virtual machine with the database image installed</Info>
  <Name>Database Instance II</Name>
  <ProductSection ovf:class="com.mydb.db">
    <Info>Specifies the OVF properties available in the OVF environment</Info>
    <Property ovf:key="ip" ovf:value="{vm2}" ovf:type="string"/>
    <Property ovf:key="ip2" ovf:value="{vm1} " ovf:type="string"/>
    <Property ovf:key="primaryAtBoot" ovf:value="no" ovf:type="string"/>
  </ProductSection>
</VirtualSystem>
```

```
<VirtualHardwareSection>
  <Info>256 MB, 1 CPU, 1 disk, 1 nic virtual machine</Info>
  <System>
    <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
    <vssd:InstanceID>0</vssd:InstanceID>
    <vssd:VirtualSystemType>vmx-04</vssd:VirtualSystemType>
  </System>
  <Item>
    <rasd:Description>Number of virtual CPUs</rasd:Description>
    <rasd:ElementName>1 virtual CPU</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
  </Item>
  <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>
  <Item>
    <rasd:AutomaticAllocation>true</rasd:AutomaticAllocation>
    <rasd:Connection>VM Network</rasd:Connection>
    <rasd:ElementName>Ethernet adapter on "VM Network"</rasd:ElementName>
    <rasd:InstanceID>3</rasd:InstanceID>
    <rasd:ResourceSubType>PCNet32</rasd:ResourceSubType>
    <rasd:ResourceType>10</rasd:ResourceType>
  </Item>
  <Item>
    <rasd:AddressOnParent>1</rasd:AddressOnParent>
    <rasd:ElementName>SCSI Controller 0 - LSI Logic</rasd:ElementName>
    <rasd:InstanceID>1000</rasd:InstanceID>
```

```
<rasd:ResourceSubType>LsiLogic</rasd:ResourceSubType>
<rasd:ResourceType>6</rasd:ResourceType>
</Item>
<Item>
  <rasd:AddressOnParent>0</rasd:AddressOnParent>
  <rasd:ElementName>Harddisk 1</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/db</rasd:HostResource>
  <rasd:InstanceID>22001</rasd:InstanceID>
  <rasd:Parent>1000</rasd:Parent>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
</VirtualHardwareSection>
<OperatingSystemSection ovf:id="97">
  <Info>Guest Operating System</Info>
  <Description>Linux 2.4.x</Description>
</OperatingSystemSection>
</VirtualSystem>
</VirtualSystemCollection>
</VirtualSystemCollection>
<!-- External I18N bundles -->
<Strings xml:lang="de-DE" ovf:fileRef="de-DE-bundle.xml"/>
<!-- Embedded I18N bundles -->
<Strings xml:lang="da-DA">
  <Msg ovf:msgid="network.description">Netværket servicen skal være tilgængelig
på</Msg>
  <Msg ovf:msgid="annotation.annotation">Kontakt kundeservice i tilfælde af
  kritiske problemer</Msg>
  <Msg ovf:msgid="property.email.description">Email adresse for administrator</Msg>
  <Msg ovf:msgid="property.appip.description">IP adresse for service</Msg>
  <Msg ovf:msgid="property.dpip">Primær IP adresse for database</Msg>
  <Msg ovf:msgid="property.dpip2.description">Sekundær IP adresse for database</Msg>
  <Msg ovf:msgid="property.loglevel.description">Logningsniveau for service</Msg>
  <Msg ovf:msgid="minimal.label">Minimal</Msg>
  <Msg ovf:msgid="minimal.description">Installer service med minimal brug af
```

```
    resourcer</Msg>
    <Msg ovf:msgid="standard.label">Normal</Msg>
    <Msg ovf:msgid="standard.description">Installer service med normal brug af
    resourcer</Msg>
  </Strings>
</Envelope>
```

## 完全な OVF 環境

Web 層および DB1 仮想マシンが見る OVF 環境を次に記載する。(DB2 は DB1 と実質的に同一なため省略した)。

### Web 層仮想マシンの OVF 環境

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:ovfenv="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  ovfenv:id="WebTier">

  <!-- Information about hypervisor platform -->
  <PlatformSection>
    <Kind>ESX Server</Kind>
    <Version>3.0.1</Version>
    <Vendor>VMware, Inc.</Vendor>
    <Locale>en_US</Locale>
  </PlatformSection>

  <!-- Properties defined for this virtual machine -->
  <PropertySection>
    <Property ovfenv:key="adminEmail" ovfenv:value="ovf-admin@vmware.com"/>
    <Property ovfenv:key="appIp" ovfenv:value="10.20.132.101"/>
    <Property ovfenv:key="dbIp" ovfenv:value="10.20.132.102"/>
    <Property ovfenv:key="db2Ip" ovfenv:value="10.20.132.103"/>
  </PropertySection>
</Environment>
```

```
<Property ovfenv:key="logLevel" ovfenv:value="warning"/>
</PropertySection>

<Entity ovfenv:id="DBTier">
  <PropertySection>
    <Property ovfenv:key="adminEmail" ovfenv:value="ovf-admin@vmware.com"/>
    <Property ovfenv:key="appIp" ovfenv:value="10.20.132.101"/>
    <Property ovfenv:key="dbIp" ovfenv:value="10.20.132.102"/>
    <Property ovfenv:key="db2Ip" ovfenv:value="10.20.132.103"/>
    <Property ovfenv:key="logLevel" ovfenv:value="warning"/>
    <Property ovfenv:key="com.mydb.db.vm1" ovfenv:value="10.20.132.102"/>
    <Property ovfenv:key="com.mydb.db.vm2" ovfenv:value="10.20.132.103"/>
    <Property ovfenv:key="com.mydb.db.log" ovfenv:value="warning"/>
  </PropertySection>
</Entity>
</Environment>
```

## DB1 仮想マシンの OVF 環境

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:ovfenv="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  ovfenv:id="DB1">

  <!-- Information about hypervisor platform -->
  <PlatformSection>
    <Kind>ESX Server</Kind>
    <Version>3.0.1</Version>
    <Vendor>VMware, Inc.</Vendor>
    <Locale>en_US</Locale>
  </PlatformSection>

  <!-- Properties defined for this virtual machine -->
```



```
<PropertySection>
  <Property ovfenv:key="com.mydb.db.vm1" ovfenv:value="10.20.132.102"/>
  <Property ovfenv:key="com.mydb.db.vm2" ovfenv:value="10.20.132.103"/>
  <Property ovfenv:key="com.mydb.db.log" ovfenv:value="warning"/>
  <Property ovfenv:key="com.mydb.db.ip" ovfenv:value="10.20.132.102"/>
  <Property ovfenv:key="com.mydb.db.ip2" ovfenv:value="10.20.132.103"/>
  <Property ovfenv:key="com.mydb.db.primaryAtBoot" ovfenv:value="yes"/>
</PropertySection>

<Entity ovfenv:id="DB2">
  <PropertySection>
    <Property ovfenv:key="com.mydb.db.vm1" ovfenv:value="10.20.132.102"/>
    <Property ovfenv:key="com.mydb.db.vm2" ovfenv:value="10.20.132.103"/>
    <Property ovfenv:key="com.mydb.db.log" ovfenv:value="warning"/>
    <Property ovfenv:key="com.mydb.db.ip" ovfenv:value="10.20.132.103"/>
    <Property ovfenv:key="com.mydb.db.ip2" ovfenv:value="10.20.132.102"/>
    <Property ovfenv:key="com.mydb.db.primaryAtBoot" ovfenv:value="no"/>
  </PropertySection>
</Entity>
</Environment>
```

## B LAMP スタック例

本例では、LAMP 仮想アプライアンスの OVF 記述子がどのようなものかに関し、具体例を 2 つ取り上げる。シングル VM LAMP 仮想アプライアンスと、複数 VM LAMP 仮想アプライアンスの両方を示す。LAMP は Linux オペレーティング・システム、Apache ウェブ・サーバー、MySQL データベース、PHP ウェブ開発ソフトウェア・パッケージを使用して構築されたサービスの略語である。

本例ではオペレーティング・システムとアプリケーション・レベル両方のデプロイメント・パラメータを指定するために、どのように **ProductSection** を使用できるかを示す。例えば、これらのパラメータは特定の環境にデプロイされたときに、サービスの性能を最適化するために使用できる。記述子は完全であるが、その他は最小限にとどめたので、例えば EULA セクションはない。

## デプロイメント時のカスタマイズ

OVF パッケージのデプロイメント段階の一部は、カスタマイズしたパラメータを提供することである。カスタマイズしたパラメータは OVF 記述子で特定され、ゲスト・ソフトウェアに OVF 環境を使用して提供される。このデプロイメント時のカスタマイズは、仮想スイッチ接続性と物理ストレージの場所を含む仮想マシン・レベルのパラメータに追加される。

LAMP ベースの仮想アプライアンスでは、デプロイメント時のカスタマイズにはサービスの IP アドレスとポート番号、ゲートウェイやサブネット等のネットワーク情報およびパラメータも含まれ、任意のデプロイメントに対して性能を最適化できる。デプロイメント担当者に明らかにされるプロパティはベンダーやサービスによって様々なである。本例の記述子では、4 つの異なる LAMP 構成要素に次のパラメータを使用する。

製品	プロパティ	記述
Linux	hostname	IP アドレスを含む、アプリケーションのネットワーク ID
	ip	
	subnet	
	gateway	
	dns	
	netCoreRmemMax netCoreWmemMax	IP スタックの転送速度を最適化するパラメータ
Apache	httpPort	ウェブ・サーバーのポート番号
	httpsPort	
	startThreads	ウェブ・サーバーのパフォーマンスを最適化するパラメータ
	minSpareThreads	
	maxSpareThreads	
	maxClients	
MySQL	queryCacheSize	データベースのパフォーマンスを最適化するパラメータ
	maxConnections	
	waitTimeout	
PHP	sessionTimeout	セッションのタイムアウト方法とセッション数を含む、PHP エンジンの動作をカスタマイズするパラメータ
	concurrentSessions	
	memoryLimit	

イタリックで書かれたパラメータはユーザーによるコンフィギュレーションが必要である。それ以外は、合理的なデフォルトを持つため、ユーザーが値を入力することが必ずしも必要ではない。

各ソフトウェア製品にあわせてカスタマイズしたパラメータは個別の製品セクションに内包される。例えば、Apache ウェブ・サーバーは、次のセクションを使用する。

```
<ProductSection ovf:class="org.apache.httpd">
  <Info>Product customization for the installed Apache Web Server</Info>
  <Product>Apache Distribution Y</Product>
  <Version>2.6.6</Version>
  <Property ovf:key="httpPort" ovf:type="uint16" ovf:value="80"
    ovf:userConfigurable="true">
    <Description>Port number for HTTP requests</Description>
  </Property>
  <Property ovf:key="httpsPort" ovf:type="uint16" ovf:value="443"
    ovf:userConfigurable="true">
    <Description>Port number for HTTPS requests</Description>
  </Property>
  <Property ovf:key="startThreads" ovf:type="uint16" ovf:value="50"
    ovf:userConfigurable="true">
    <Description>Number of threads created on startup. </Description>
  </Property>
  <Property ovf:key="minSpareThreads" ovf:type="uint16" ovf:value="15"
    ovf:userConfigurable="true">
    <Description> Minimum number of idle threads to handle request spikes.</Description>
  </Property>
  <Property ovf:key="maxSpareThreads" ovf:type="uint16" ovf:value="30"
    ovf:userConfigurable="true">
    <Description>Maximum number of idle threads </Description>
  </Property>
  <Property ovf:key="maxClients" ovf:type="uint16" ovf:value="256"
    ovf:userConfigurable="true">
```

```
<Description>Limit the number of simultaneous requests that will be served.
</Description>
  </Property>
</ProductSection>
```

ovf:class="org.apache.httpd"属性はプロパティのプレフィクスを指定する。従って、Apache データベースは OVF 環境の次のプロパティを探すことが期待される。

```
<Environment
...
<!-- Properties defined for this virtual machine -->
<PropertySection>
  <Property ovfenv:name="org.apache.httpd.httpPort ovfenv:value="80"/>
  <Property ovfenv:name="org.apache.httpd.httpsPort ovfenv:value="443"/>
  <Property ovfenv:name="org.apache.httpd.startThreads" ovfenv:value="50"/>
  <Property ovfenv:name="org.apache.httpd.minSpareThreads" ovfenv:value="15"/>
  <Property ovfenv:name="org.apache.httpd.maxSpareThreads" ovfenv:value="30"/>
  <Property ovfenv:name="org.apache.httpd.maxClients" ovfenv:value="256"/>
  ...
</PropertySection>
...
</Environment>
```

## シンプル LAMP OVF 記述子

LAMP スタックを持つシングル VM 仮想アプライアンスの完全な OVF 記述子を次に記載する。

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope
  xmlns="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
  ta"
```

```
xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"

<!-- References to all external files -->
<References>
  <File ovf:id="lamp" ovf:href="lamp.vmdk" ovf:size="180114671"/>
</References>

<!-- Describes meta-information about all virtual disks in the package. -->
<DiskSection>
  <Info>List of the virtual disks used in the package</Info>
  <Disk ovf:diskId="lamp" ovf:fileRef="lamp" ovf:capacity="4294967296"
    ovf:populatedSize="1924967692"
    ovf:format="http://www.vmware.com/specifications/vmdk.html#streamOptimized"/>
</DiskSection>

<!-- Describes all networks used in the package -->
<NetworkSection>
  <Info>Logical networks used in the package</Info>
  <Network ovf:name="VM Network">
    <Description>The network that the LAMP Service will be available
    on</Description>
  </Network>
</NetworkSection>

<VirtualSystem ovf:id="MyLampService">
  <Info>Single-VM Virtual appliance with LAMP stack</Info>
  <Name>LAMP Virtual Appliance</Name>
  <!-- Overall information about the product -->
  <ProductSection>
    <Info>Product information for the service</Info>
    <Product>Lamp Service</Product>
    <Version>1.0</Version>
    <FullVersion>1.0.0</FullVersion>
  </ProductSection>
  <!-- Linux component configuration parameters -->
  <ProductSection ovf:class="org.linuxdistx">
    <Info>Product customization for the installed Linux system</Info>
```

```
<Product>Linux Distribution X</Product>
<Version>2.6.3</Version>
<Property ovf:key="hostname" ovf:type="string">
  <Description>Specifies the hostname for the appliance</Description>
</Property>
<Property ovf:key="ip" ovf:type="string">
  <Description>Specifies the IP address for the appliance</Description>
</Property>
<Property ovf:key="subnet" ovf:type="string">
  <Description> Specifies the subnet to use on the deployed network
  </Description>
</Property>
<Property ovf:key="gateway" ovf:type="string">
  <Description> Specifies the gateway on the deployed network
  </Description>
</Property>
<Property ovf:key="dns" ovf:type="string">
  <Description> A comma separated list of DNS servers on the deployed
  network </Description>
</Property>
<Property ovf:key="netCoreRmemMaxMB" ovf:type="uint16" ovf:value="16"
  ovf:userConfigurable="true">
  <Description> Specify TCP read max buffer size in mega bytes. Default is
  16. </Description>
</Property>
<Property ovf:key="netCoreWmemMaxMB" ovf:type="uint16" ovf:value="16"
  ovf:userConfigurable="true">
  <Description> Specify TCP write max buffer size in mega bytes. Default is
  16. </Description>
</Property>
</ProductSection>
<!-- Apache component configuration parameters -->
<ProductSection ovf:class="org.apache.httpd">
  <Info>Product customization for the installed Apache Web Server</Info>
```

```
<Product>Apache Distribution Y</Product>
<Version>2.6.6</Version>
<Property ovf:key="httpPort" ovf:type="uint16" ovf:value="80"
  ovf:userConfigurable="true">
  <Description>Port number for HTTP requests</Description>
</Property>
<Property ovf:key="httpsPort" ovf:type="uint16" ovf:value="443"
  ovf:userConfigurable="true">
  <Description>Port number for HTTPS requests</Description>
</Property>
<Property ovf:key="startThreads" ovf:type="uint16" ovf:value="50"
  ovf:userConfigurable="true">
  <Description>Number of threads created on startup. </Description>
</Property>
<Property ovf:key="minSpareThreads" ovf:type="uint16" ovf:value="15"
  ovf:userConfigurable="true">
  <Description> Minimum number of idle threads to handle request spikes.
  </Description>
</Property>
<Property ovf:key="maxSpareThreads" ovf:type="uint16" ovf:value="30"
  ovf:userConfigurable="true">
  <Description>Maximum number of idle threads </Description>
</Property>
<Property ovf:key="maxClients" ovf:type="uint16" ovf:value="256"
  ovf:userConfigurable="true">
  <Description>Limit the number of simultaneous requests that will be
  served. </Description>
</Property>
</ProductSection>
<!-- MySQL component configuration parameters -->
<ProductSection ovf:class="org.mysql.db">
  <Info>Product customization for the installed MySql Database Server</Info>
  <Product>MySQL Distribution Z</Product>
  <Version>5.0</Version>
```

```
<Property ovf:key="queryCacheSizeMB" ovf:type="uint16" ovf:value="32"
  ovf:userConfigurable="true">
  <Description>Buffer to cache repeated queries for faster access (in
  MB)</Description>
</Property>

<Property ovf:key="maxConnections" ovf:type="uint16" ovf:value="500"
  ovf:userConfigurable="true">
  <Description>The number of concurrent connections that can be
  served</Description>
</Property>

<Property ovf:key="waitTimeout" ovf:type="uint16" ovf:value="100"
  ovf:userConfigurable="true">
  <Description>Number of seconds to wait before timing out a connection
  </Description>
</Property>
</ProductSection>

<!-- PHP component configuration parameters -->
<ProductSection ovf:class="net.php">
  <Info>Product customization for the installed PHP component</Info>
  <Product>PHP Distribution U</Product>
  <Version>5.0</Version>
  <Property ovf:key="sessionTimeout" ovf:type="uint16" ovf:value="5"
    ovf:userConfigurable="true">
    <Description> How many minutes a session has to be idle before it is
    timed out </Description>
  </Property>
  <Property ovf:key="concurrentSessions" ovf:type="uint16" ovf:value="500"
    ovf:userConfigurable="true">
    <Description> The number of concurrent sessions that can be served
    </Description>
  </Property>
  <Property ovf:key="memoryLimit" ovf:type="uint16" ovf:value="32"
    ovf:userConfigurable="true">
    <Description> How much memory in megabytes a script can consume before
```



```
        being killed </Description>
    </Property>
</ProductSection>
<OperatingSystemSection ovf:id="99">
    <Info>Guest Operating System</Info>
    <Description>Linux 2.6.x</Description>
</OperatingSystemSection>
<VirtualHardwareSection>
    <Info>Virtual Hardware Requirements: 256MB, 1 CPU, 1 disk, 1 NIC</Info>
    <System>
        <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
        <vssd:InstanceID>0</vssd:InstanceID>
        <vssd:VirtualSystemType>vmx-04</vssd:VirtualSystemType>
    </System>
    <Item>
        <rasd:Description>Number of virtual CPUs</rasd:Description>
        <rasd:ElementName>1 virtual CPU</rasd:ElementName>
        <rasd:InstanceID>1</rasd:InstanceID>
        <rasd:ResourceType>3</rasd:ResourceType>
        <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
    </Item>
    <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>
    <Item>
        <rasd:AutomaticAllocation>true</rasd:AutomaticAllocation>
        <rasd:Connection>VM Network</rasd:Connection>
        <rasd:ElementName>Ethernet adapter on "VM Network"</rasd:ElementName>
        <rasd:InstanceID>3</rasd:InstanceID>
```

```
<rasd:ResourceType>10</rasd:ResourceType>
</Item>
<Item>
  <rasd:ElementName>SCSI Controller 0 - LSI Logic</rasd:ElementName>
  <rasd:InstanceID>4</rasd:InstanceID>
  <rasd:ResourceSubType>LsiLogic</rasd:ResourceSubType>
  <rasd:ResourceType>6</rasd:ResourceType>
</Item>
<Item>
  <rasd:ElementName>Harddisk 1</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/lamp</rasd:HostResource>
  <rasd:InstanceID>5</rasd:InstanceID>
  <rasd:Parent>4</rasd:Parent>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
</VirtualHardwareSection>
</VirtualSystem>
</Envelope>
```

## 2 層 LAMP OVF 記述子

2 層 LAMP スタックでは、アプリケーション層（Linux、Apache、PHP）とデータベース層（Linux、MySQL サーバー）は、広い拡張性を保つために別々の仮想マシンとして動作する。

OVF フォーマットによって、サービスを実装する方法がユーザーにはほとんど気付かれないうようになる。特に、シングル VM または 2 層 LAMP アプライアンスをインストールする際のデプロイメント・エクスペリエンスはほとんど同じである。はっきりした違いは、ユーザーは IP アドレスを 2 つと DNS ホスト名を 2 つ提供する必要があるということだけである。

シングル VM 記述子と比べると、次のような変更点がある。

ユーザー設定ができるパラメータはすべて、VirtualSystemCollection エンティティに置かなければならない。Apache、MySQL、PHP の ProductSection はシングル VM の例と同じである。

2つの仮想マシンのLinuxソフトウェアの設定は、ほとんど同じパラメータを使うが、多少異なる（IPとホスト名）。新しいProductSectionがVirtualSystemCollectionに追加され、ユーザーにプロンプト表示する。\${property}表現を使用して各VirtualSystemエンティティで値を割り当てる。

ディスク・チェーンを使用してシングルVMアプライアンスと同程度のサイズに抑えてダウンロードする。Linuxのインストールは共有ベースディスクに保存されるため、ダウンロードするLinuxのコピーは効率的に1つですむ。

完全なOVF記述子を次に示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope
  xmlns="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  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"
  <!-- References to all external files. -->
  <References>
    <File ovf:id="lamp-base" ovf:href="lampdb.vmdk" ovf:size="180114671"/>
    <File ovf:id="lamp-db" ovf:href="lampdb.vmdk" ovf:size="1801146"/>
    <File ovf:id="lamp-app" ovf:href="lampapp.vmdk" ovf:size="34311371"/>
  </References>
  <!-- Describes meta-information about all virtual disks in the package.
  This example is encoded as a delta-disk hierarchy.
  -->
  <DiskSection>
    <Info>List of the virtual disks used in the package</Info>
    <Disk ovf:diskId="lamp-base" ovf:fileRef="lamp-base" ovf:capacity="4294967296"
      ovf:populatedSize="1924967692"
      ovf:format="http://www.vmware.com/specifications/vmdk.html#streamOptimized"/>
```

```
<Disk ovf:diskId="lamp-db" ovf:fileRef="lamp-db" ovf:capacity="4294967296"
  ovf:populatedSize="19249672"
  ovf:format="http://www.vmware.com/specifications/vmdk.html#streamOptimized"
  ovf:parentRef="lamp-base"/>
<Disk ovf:diskId="lamp-app" ovf:fileRef="lamp-app" ovf:capacity="4294967296"
  ovf:populatedSize="2349692"
  ovf:format="http://www.vmware.com/specifications/vmdk.html#streamOptimized"
  ovf:parentRef="lamp-base"/>
</DiskSection>
<!-- Describes all networks used in the package -->
<NetworkSection>
  <Info>Logical networks used in the package</Info>
  <Network ovf:name="VM Network">
    <Description>The network that the LAMP Service will be available
    on</Description>
  </Network>
</NetworkSection>
<VirtualSystemCollection ovf:id="LampService">
  <Info>Virtual appliance with a 2-tier distributed LAMP stack</Info>
  <Name>LAMP Service</Name>
  <!-- Overall information about the product -->
  <ProductSection>
    <Info>Product information for the service</Info>
    <Product>My Lamp Service</Product>
    <Version>1.0</Version>
    <FullVersion>1.0.0</FullVersion>
  </ProductSection>
  <ProductSection>
    <Info>Product customization for Operating System Level</Info>
    <Product>Linux Distribution X</Product>
    <Version>2.6.3</Version>
    <Property ovf:key="dbHostname" ovf:type="string">
      <Description>Specifies the hostname for database virtual
      machine</Description>
    </Property>
  </ProductSection>
</VirtualSystemCollection>
```

```
</Property>
<Property ovf:key="appHostname" ovf:type="string">
  <Description>Specifies the hostname for application server virtual
    machine</Description>
</Property>
<Property ovf:key="dbIp" ovf:type="string">
  <Description>Specifies the IP address for the database virtual
    machine</Description>
</Property>
<Property ovf:key="appIp" ovf:type="string">
  <Description>Specifies the IP address for application server
    VM</Description>
</Property>
<Property ovf:key="subnet" ovf:type="string">
  <Description> Specifies the subnet to use on the deployed network
  </Description>
</Property>
<Property ovf:key="gateway" ovf:type="string">
  <Description> Specifies the gateway on the deployed network
  </Description>
</Property>
<Property ovf:key="dns" ovf:type="string">
  <Description> A comma separated list of DNS servers on the deployed
    network </Description>
</Property>
<Property ovf:key="netCoreRmemMaxMB" ovf:type="uint16" ovf:value="16"
  ovf:userConfigurable="true">
  <Description> Specify TCP read max buffer size in mega bytes. Default is
    16. </Description>
</Property>
<Property ovf:key="netCoreWmemMaxMB" ovf:type="uint16" ovf:value="16"
  ovf:userConfigurable="true">
  <Description> Specify TCP write max buffer size in mega bytes. Default is
    16. </Description>
```

```
</Property>
</ProductSection>
<!-- Apache component configuration parameters -->
<ProductSection ovf:class="org.apache.httpd">
  <Info>Product customization for the installed Apache Web Server</Info>
  <Product>Apache Distribution Y</Product>
  <Version>2.6.6</Version>
  <Property ovf:key="httpPort" ovf:type="uint16" ovf:value="80"
    ovf:userConfigurable="true">
    <Description>Port number for HTTP requests</Description>
  </Property>
  <Property ovf:key="httpsPort" ovf:type="uint16" ovf:value="443"
    ovf:userConfigurable="true">
    <Description>Port number for HTTPS requests</Description>
  </Property>
  <Property ovf:key="startThreads" ovf:type="uint16" ovf:value="50"
    ovf:userConfigurable="true">
    <Description>Number of threads created on startup. </Description>
  </Property>
  <Property ovf:key="minSpareThreads" ovf:type="uint16" ovf:value="15"
    ovf:userConfigurable="true">
    <Description>Minimum number of idle threads to handle request spikes.
    </Description>
  </Property>
  <Property ovf:key="maxSpareThreads" ovf:type="uint16" ovf:value="30"
    ovf:userConfigurable="true">
    <Description>Maximum number of idle threads </Description>
  </Property>
  <Property ovf:key="maxClients" ovf:type="uint16" ovf:value="256"
    ovf:userConfigurable="true">
    <Description>Limits the number of simultaneous requests that will be
    served. </Description>
  </Property>
</ProductSection>
```

```
<!-- MySQL component configuration parameters -->
<ProductSection ovf:class="org.mysql.db">
  <Info>Product customization for the installed MySql Database Server</Info>
  <Product>MySQL Distribution Z</Product>
  <Version>5.0</Version>
  <Property ovf:key="queryCacheSizeMB" ovf:type="uint16" ovf:value="32"
    ovf:userConfigurable="true">
    <Description>Buffer to cache repeated queries for faster access (in
    MB)</Description>
  </Property>
  <Property ovf:key="maxConnections" ovf:type="uint16" ovf:value="500"
    ovf:userConfigurable="true">
    <Description>The number of concurrent connections that can be
    served</Description>
  </Property>
  <Property ovf:key="waitTimeout" ovf:type="uint16" ovf:value="100"
    ovf:userConfigurable="true">
    <Description>Number of seconds to wait before timing out a connection
    </Description>
  </Property>
</ProductSection>
<!-- PHP component configuration parameters -->
<ProductSection ovf:class="net.php">
  <Info>Product customization for the installed PHP component</Info>
  <Product>PHP Distribution U</Product>
  <Version>5.0</Version>
  <Property ovf:key="sessionTimeout" ovf:type="uint16" ovf:value="5"
    ovf:userConfigurable="true">
    <Description> How many minutes a session has to be idle before it is
    timed out </Description>
  </Property>
  <Property ovf:key="concurrentSessions" ovf:type="uint16" ovf:value="500"
    ovf:userConfigurable="true">
    <Description> The number of concurrent sessions that can be served
```

```
    </Description>
  </Property>
  <Property ovf:key="memoryLimit" ovf:type="uint16" ovf:value="32"
    ovf:userConfigurable="true">
    <Description> How much memory in megabytes a script can consume before
      being killed </Description>
  </Property>
</ProductSection>
<StartupSection>
  <Info>Startup order of the virtual machines</Info>
  <Item ovf:id="DbServer" ovf:order="1" ovf:startDelay="120"
    ovf:startAction="powerOn" ovf:waitingForGuest="true" ovf:stopDelay="120"
    ovf:stopAction="guestShutdown"/>
  <Item ovf:id="AppServer" ovf:order="2" ovf:startDelay="120"
    ovf:startAction="powerOn" ovf:waitingForGuest="true" ovf:stopDelay="120"
    ovf:stopAction="guestShutdown"/>
</StartupSection>
<VirtualSystem ovf:id="AppServer">
  <Info>The configuration of the AppServer virtual machine</Info>
  <Name>Application Server</Name>
  <!-- Linux component configuration parameters -->
  <ProductSection ovf:class="org.linuxdistx">
    <Info>Product customization for the installed Linux system</Info>
    <Product>Linux Distribution X</Product>
    <Version>2.6.3</Version>
    <Property ovf:key="hostname" ovf:type="string" ovf:value="{appHostName}"/>
    <Property ovf:key="ip" ovf:type="string" ovf:value="{appIp}"/>
    <Property ovf:key="subnet" ovf:type="string" ovf:value="{subnet}"/>
    <Property ovf:key="gateway" ovf:type="string" ovf:value="{gateway}"/>
    <Property ovf:key="dns" ovf:type="string" ovf:value="{dns}"/>
    <Property ovf:key="netCoreRmemMaxMB" ovf:type="string"
      ovf:value="{netCoreRmemMaxMB}"/>
    <Property ovf:key="netCoreWmemMaxMB" ovf:type="string"
      ovf:value="{netCoreWmemMaxMB}"/>
  </ProductSection>
</VirtualSystem>
</ProductSection>
</ProductSection>
```



```
</ProductSection>
<OperatingSystemSection ovf:id="99">
  <Info>Guest Operating System</Info>
  <Description>Linux 2.6.x</Description>
</OperatingSystemSection>
<VirtualHardwareSection>
  <Info>Virtual Hardware Requirements: 256 MB, 1 CPU, 1 disk, 1 NIC</Info>
  <System>
    <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
    <vssd:InstanceID>0</vssd:InstanceID>
    <vssd:VirtualSystemType>vmx-04</vssd:VirtualSystemType>
  </System>
  <Item>
    <rasd:Description>Number of virtual CPUs</rasd:Description>
    <rasd:ElementName>1 virtual CPU</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
  </Item>
  <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>
  <Item>
    <rasd:AutomaticAllocation>true</rasd:AutomaticAllocation>
    <rasd:Connection>VM Network</rasd:Connection>
    <rasd:ElementName>Ethernet adapter on "VM Network"</rasd:ElementName>
    <rasd:InstanceID>3</rasd:InstanceID>
    <rasd:ResourceSubType>PCNet32</rasd:ResourceSubType>
    <rasd:ResourceType>10</rasd:ResourceType>
```

```
</Item>
<Item>
  <rasd:ElementName>SCSI Controller 0 - LSI Logic</rasd:ElementName>
  <rasd:InstanceID>4</rasd:InstanceID>
  <rasd:ResourceSubType>LsiLogic</rasd:ResourceSubType>
  <rasd:ResourceType>6</rasd:ResourceType>
</Item>
<Item>
  <rasd:ElementName>Harddisk 1</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/lamp-app</rasd:HostResource>
  <rasd:InstanceID>5</rasd:InstanceID>
  <rasd:Parent>4</rasd:Parent>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
</VirtualHardwareSection>
</VirtualSystem>
<VirtualSystem ovf:id="DB Server">
  <Info>The configuration of the database virtual machine</Info>
  <Name>Database Server</Name>
  <!-- Linux component configuration parameters -->
  <ProductSection ovf:class="org.linuxdistx">
    <Info>Product customization for the installed Linux system</Info>
    <Product>Linux Distribution X</Product>
    <Version>2.6.3</Version>
    <Property ovf:key="hostname" ovf:type="string"
      ovf:value="{dbHostName}"/>
    <Property ovf:key="ip" ovf:type="string" ovf:value="{dbIp}"/>
    <Property ovf:key="subnet" ovf:type="string" ovf:value="{subnet}"/>
    <Property ovf:key="gateway" ovf:type="string" ovf:value="{gateway}"/>
    <Property ovf:key="dns" ovf:type="string" ovf:value="{dns}"/>
    <Property ovf:key="netCoreRmemMaxMB" ovf:type="string"
      ovf:value="{netCoreRmemMaxMB}"/>
    <Property ovf:key="netCoreWmemMaxMB" ovf:type="string"
      ovf:value="{netCoreWmemMaxMB}"/>
  </ProductSection>
</VirtualSystem>
```

```
</ProductSection>
<OperatingSystemSection ovf:id="99">
  <Info>Guest Operating System</Info>
  <Description>Linux 2.6.x</Description>
</OperatingSystemSection>
<VirtualHardwareSection>
  <Info>Virtual Hardware Requirements: 256 MB, 1 CPU, 1 disk, 1 nic</Info>
  <System>
    <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
    <vssd:InstanceID>0</vssd:InstanceID>
    <vssd:VirtualSystemType>vmx-04</vssd:VirtualSystemType>
  </System>
  <Item>
    <rasd:Description>Number of virtual CPUs</rasd:Description>
    <rasd:ElementName>1 virtual CPU</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
  </Item>
  <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>
  <Item>
    <rasd:AutomaticAllocation>true</rasd:AutomaticAllocation>
    <rasd:Connection>VM Network</rasd:Connection>
    <rasd:ElementName>Ethernet adapter on "VM Network"</rasd:ElementName>
    <rasd:InstanceID>3</rasd:InstanceID>
    <rasd:ResourceType>10</rasd:ResourceType>
  </Item>
</VirtualHardwareSection>
```

```
<Item>
  <rasd:ElementName>SCSI Controller 0 - LSI Logic</rasd:ElementName>
  <rasd:InstanceID>4</rasd:InstanceID>
  <rasd:ResourceSubType>LsiLogic</rasd:ResourceSubType>
  <rasd:ResourceType>6</rasd:ResourceType>
</Item>
<Item>
  <rasd:ElementName>Harddisk 1</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/lamp-db</rasd:HostResource>
  <rasd:InstanceID>5</rasd:InstanceID>
  <rasd:Parent>4</rasd:Parent>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
</VirtualHardwareSection>
</VirtualSystem>
</VirtualSystemCollection>
</Envelope>
```

## C 拡張性の例

OVF 仕様を使うことによって、カスタム・メタデータを OVF 記述子に複数の方法で追加できる。

新しいセクション・エレメントを **Section** 置換グループの一部として定義し、OVF スキーマがセクションに存在することを許可する場所ではどこでも使うことができる。

OVF スキーマはオープン・コンテンツ・モデルを使用し、すべての既存タイプは追加エレメントを末尾につけて拡張できる。拡張ポイントは OVF スキーマに `namespace="##other"` を持つ `xs:any` 宣言と共に宣言されている。

OVF スキーマは既存のタイプ上に追加の属性を許可する。

カスタム・メタデータは OVF XML 名前空間を使用できない。カスタム・エレメント上では、ブーリアン `ovf:required` 属性が、エレメント内の情報が正しい動作に必須であるか、または任意かを指定する。

OVF スキーマのオープン・コンテンツ・モデルは既存のタイプの末尾を拡張することのみ

許可する。XML スキーマ 1.0 を使うと、ユニーク・パーティクル属性規則と `xs:any` 宣言をスキーマ全体に追加する必要があるため、より柔軟なオープン・コンテンツ・モデルを許可することは難しい。XML スキーマ 1.1 標準案は、`xs:openContent mode="interleave"` 宣言を使用する、はるかに柔軟なオープン・コンテンツ・メカニズムを持つ。OVF 仕様の将来のバージョンではこの点のサポートを取り扱う。

## カスタム・スキーマ

2 つの拡張タイプを定義するカスタム XML スキーマを次に記載する。最初の宣言は OVF Section 置換グループのカスタム・メンバを定義し、二番目の宣言はシンプル・カスタム・タイプを定義する。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="http://schemas.customextension.org/1"
  xmlns:custom="http://schemas.customextension.org/1"
  xmlns="http://schemas.customextension.org/1"
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  attributeFormDefault="qualified"
  elementFormDefault="qualified">

  <!-- Define a custom member of the ovf:Section substitution group -->
  <xs:element name="CustomSection" type="custom:CustomSection_Type"
substitutionGroup="ovf:Section"/>

  <xs:complexType name="CustomSection_Type">
    <xs:complexContent>
      <xs:extension base="ovf:Section_Type">
        <xs:sequence>
          <xs:element name="Data" type="xs:string"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
<!-- Define other simple custom type not part of ovf:Section substitution group -->
<xs:complexType name="CustomOther_Type">
  <xs:sequence>
    <xs:element name="Data" type="xs:string"/>
  </xs:sequence>
  <xs:attribute ref="ovf:required"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
</xs:schema >
```

### カスタム拡張を持つ記述子

前記のカスタム・スキーマを使用する完全な OVF 記述子を次に記載する。記述子は OVF スキーマおよびカスタム・スキーマに対して検証を行うが、拡張例以外にも、記述子は最小限であり、従ってあまり役に立たない。

記述子は 3 種類すべての拡張子タイプを持つ。カスタム OVF Section エレメント、拡張ポイントでのカスタム・エレメント、そしてカスタム属性である。

```
<?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"
  xmlns:custom="http://schemas.customextension.org/1">
  <!-- Dummy References element -->
  <References/>
```

```
<!-- EXAMPLE: Optional custom OVF section element with validation against custom schema
-->
<custom:CustomSection ovf:required="false">
  <Info>Description of custom extension</Info>
  <custom:Data>somevalue</custom:Data>
</custom:CustomSection>

<!-- Describes all networks used in the package -->
<NetworkSection>
  <Info>Logical networks used in the package</Info>
  <!-- EXAMPLE: Optional custom attribute -->
  <Network ovf:name="VM Network" custom:desiredCapacity="1 Gbit/s"/>
  <!-- EXAMPLE: Optional custom meta-data inserted at extension point with validation
        against custom schema -->
  <custom:CustomOther xsi:type="custom:CustomOther_Type" ovf:required="false">
    <custom:Data>somevalue</custom:Data>
  </custom:CustomOther>
</NetworkSection>

<!-- Dummy Content element -->
<VirtualSystem ovf:id="Dummy">
  <Info>Dummy VirtualSystem</Info>
</VirtualSystem>
</Envelope>
```

OVF 環境の XML スキーマは、OVF エンベロープ XML スキーマの拡張メカニズムと一致する拡張メカニズムを持つため、OVF 環境文書は同様に拡張可能である。