



文件编号：DSP2017

日期：2014年4月24日

版本：2.0.0

## 开放虚拟化格式白皮书

文件类型：DMTF 供参考文件

文件状态：DMTF 供参考文件

文件语言：中文（中国）

版权通知

著作权©2007-2017，分布式管理任务组有限公司（DMTF），保留所有权利。

DMTF 是一家非营利性组织，其行业成员的主要目的在于促进企业和系统的管理与互通性。成员和非成员都可以复制 DMTF 的规范和文件，但在引用相关文件时必须明确说明其具体出处。DMTF 规范文件可能会随时进行修改，因此用户必须随时注意相关文件的具体版本号和发行日期。

实施本标准或建议标准中的部分元素可能会涉及到第三方的专利权，包括临时性专利权（简称为“专利权”）。DMTF 不负责向标准用户告知或说明上述专利权，也不负责识别、披露。或者确定任何或所有上述第三方专利权、所有者或索赔人；也不对在确定或者披露上述权利、所有者或索赔人过程中出现的不完整或不准确信息负责。在任何情况下，DMTF 都不对未能识别、披露或者确定上述第三方专利权的行爲、也不对相关方依赖标准文件，或者将其纳入到其产品、协议或测试流程中的行爲承担任何形式的责任。DMTF 不对实施本标准文件的任何一方承担责任，不论相关实施行爲是否可以被预见；也不对任何专利所有者或索赔人承担责任；也不对标准在出版之后被撤销或者进行修改导致的任何成本或损失承担责任。对于实施本标准的任何方而言，如果其在实施标准的过程中侵犯了任何专利权，则其必须针对专利所有者提出的索赔要求向 DMTF 进行赔偿，并保证 DMTF 免受任何不利影响。

更多关于向 DMTF 发出通知，认为其持有的专利权可能与 DMTF 标准相关、或者会影响标准实施的持有专利权的第三方的信息，请参阅下列网站中的相关内容：<http://www.dmtf.org/about/policies/disclosures.php>。

## 摘要

“开放虚拟化格式（OVF）白皮书”中描述了 DSP 0243、DSP 8023 和 DSP 8027 等规范的具体应用，以及构成开放虚拟化格式（OVF）标准一部分的相关规范。本文件的预计受众是希望理解 OVF 包以及其在具体用途实例中的应用的任何人。本文件假设受众熟悉虚拟化，且了解 CIM 模型的一些基本概念。

## 目录

1. 概述.....	7
1.1 概述.....	7
1.2 设计理念.....	8
2. OVF 核心概念.....	9
2.1 虚拟应用.....	9
2.2 寿命周期.....	10
3. XML 后台.....	12
3.1 XML 架构的使用.....	12
3.2 在 OVF 中使用的基本 XML 概念.....	13
3.2.1 元素.....	13
3.2.2 属性.....	13
3.2.3 置换群.....	13
4. OVF 包.....	14
4.1 OVF 描述文件结构.....	14
4.2 OVF 架构中规定的全局属性.....	15
4.3 OVF 可扩展性.....	15
4.4 OVF 顶层元素.....	17
4.4.1 VirtualSystem 元素.....	20
4.4.2 VirtualSystemCollection 元素.....	20
4.4.3 References 元素.....	20
4.4.4 DiskSection 元素.....	20
4.4.5 NetworkSection 元素.....	21
4.4.6 DeploymentOptionsSection 元素.....	24
4.4.7 SharedDiskSection 元素.....	28
4.4.8 PlacementGroupSection 元素.....	30
4.5 在虚拟系统和虚拟系统集合中使用的 OVF 字段元素.....	31
4.5.1 AnnotationSection 元素.....	31
4.5.2 ProductSection 元素.....	31
4.5.3 EulaSection 元素.....	32
4.5.4 VirtualHardwareSection 元素.....	33
4.5.5 PlacementSection 元素.....	35
4.5.6 EncryptionSection 元素.....	39
4.6 虚拟系统集合中使用的 OVF 字段元素.....	43
4.6.1 ResourceAllocationSection 元素.....	43
4.6.2 StartupSection 元素.....	44
4.6.3 ScaleOutSection 元素.....	45
4.7 虚拟系统中使用的 OVF 字段元素.....	47
4.7.1 OperatingSystemSection 元素.....	47
4.7.2 InstallSection 元素.....	48
4.7.3 EnvironmentFilesSection 元素.....	48
4.7.4 BootDeviceSection 元素.....	49
5. OVF 包授权.....	50
5.1 创建.....	50

5.2 国际化 .....	51
5.3 可扩展性 .....	53
5.3.1 置换群 .....	54
5.3.2 元素 .....	56
5.3.3 属性 .....	56
5.4 符合性 .....	57
5.5 虚拟硬件描述 .....	57
5.6 描述文件示例 .....	59
6. OVF 包部署 .....	59
6.1 部署 .....	59
6.2 OVF 环境描述文件 .....	60
6.3 部署过程中的资源配置选项 .....	60
6.4 属性元素部署过程中的产品自定义 .....	62
7. 可移植性 .....	65
附件 A (供参考): 单一虚拟系统示例 .....	67
附件 B (供参考): 多层次 PetStore 示例 .....	70
B.1 架构和打包 .....	70
B.2 属性 .....	70
B.3 磁盘布置 .....	72
B.4 Pet Store OVF 描述文件 .....	72
B.5 完整 OVF 环境 .....	81
附件 C (供参考): 单一虚拟系统 LAMP 堆栈示例 .....	84
C.1 部署时自定义 .....	84
C.2 简单 LAMP OVF 描述文件 .....	86
附件 D (供参考): 多重虚拟系统 LAMP 堆栈示例 .....	92
D.1 双层 LAMP OVF 描述文件 .....	92
附件 E (供参考): 可扩展性示例 .....	100
E.1 自定义架构 .....	100
E.2 带有自定义扩展项的描述文件 .....	101
附件 F (供参考): 修订记录 .....	103

## 插图目录

图 1: OVF 包寿命周期 .....	11
图 2: OVF 授权功能 .....	11
图 4: OVF 包结构 .....	15
图 5: 网络连接 .....	23
图 6: LAN-SAN 网络连接 .....	23
图 7: 亲和性布置 .....	36
图 8: 可用性布置 .....	37
图 9: 亲和性和可用性布置 .....	39
图 10: StartupSection (启动字段) 遍历 .....	44
图 B-1: Pet Store OVF 包 .....	70
图 B-2: Pet Store 虚拟磁盘布置 .....	72

## 表格目录

表 B-1: 网页层面配置 .....	71
表 B-2: 数据库层面配置 .....	71
表 C-1: LAMP 配置 .....	84

## 序言

“开放虚拟化格式（OVF）白皮书”（DSP 2017）是由 DMTF 的 OVF 工作小组编制的。

本 DMTF 参考性规范是很多个人和团队共同努力的结果，其中包括：

劳伦斯.拉莫斯：威睿公司（主编）；

马尔文.瓦斯奇科：DMTF 人员（合作编辑）；

皮特.沃恩德勒：爱立信公司（合作编辑）；

埃里克.威尔士：日立公司（合作编辑）；

何马尔.沙哈：博通公司；

施什尔.帕迪卡：思杰系统公司；

理查德.蓝道尔：DMTF 人员；

罗伯特.弗洛伊德：日立公司；

杰夫.惠尔勒：华为公司；

莫妮卡.马丁：微软公司；

程伟：微软公司；

斯瑞尼瓦斯.马杜里：甲骨文公司；

史蒂芬.格拉鲁普：威睿公司；

雷尼.施密特：威睿公司；

嘉珊法尔.阿里：中兴通讯公司。

# 开放虚拟化格式白皮书

## 1. 概述

### 1.1 概述

“开放虚拟化格式（OVF）规范”中将为行业提供基于虚拟系统的标准的软件解决方案的打包格式，能够解决软件供货商和云计算服务提供商的重要业务需求。

OVF 包可以供独立软件供货商（ISV）用于发布软件解决方案；可以供数据中心操作员用于将软件解决方案从一个数据中心转移到另一个；可以供客户用于归档软件解决方案；也可以用于其他具有标准化软件解决方案打包需求的用途。

下列用途实例是 OVF 包的主要基础：

- 1) OVF 包能够帮助独立软件供货商打包软件解决方案，使其能够供多个管理程序使用；
- 2) OVF 包能够打包虚拟系统或者虚拟系统集合，确保能够将其从一个数据中心转移到另一个处。

另外，本规范还适用于其他用途实例和衍生用途实例（即：子集）。

OVF 第一版被行业内普遍采纳，现在已经成为了国际标准。

OVF 第二版中加入了一些强化的打包功能，使其适用于更加广泛的用途实例，可以在诸如云计算领域等新兴行业中得到利用。

OVF 2 中加入了下列功能：

- 支持网络端口；
- 可以按照部署时间缩放；
- 支持基本布局政策；
- OVF 包加密处理；
- 运行时磁盘共享；
- 高级设备引导顺序；
- 向客户端软件转移高级数据；
- 支持改进后的国际化-I18N；
- 支持改进后的 HASH；
- 更新了 CIM 架构。

在适用的情况下，OVF 采用了通用信息模型（CIM），以确保管理软件能够使用开放式标准，清晰地理解和更便捷地映射资源属性。使用具体设备的 CIM\_ResrouceAllocationSettingData（CIM\_资源分配设置数据）分级和子分级来规定虚拟系统需要运行的具体资源。

OVF2 支持网络配置，支持使用“网络端口配置（DSP 8049）”的 IEEE 边缘虚拟桥接发现和配置协议。重要的属性将在 CIM\_EthernetPortAllocationSettingData（CIM\_以太网端口分配设置数据）中提供。

本文件的目标在于提供 OVF 动机、目标、设计和预计用途方面的详细信息，本文件应该与相同主要版本的“OVF 规范”结合使用。

## 1.2 设计理念

随着虚拟基础设施的快速发展，需要提供一个标准化、便携式元数据格式，以便在虚拟平台上或平台之间进行虚拟系统的分配。因此，将软件应用和对其进行认证的操作系统打包在一起，形成一个可以容易被独立软件供货商转移的格式，之后通过测试、开发和生产，在不依赖外部机构的情况下形成一个预先配置、预先打包的单元的能力就非常富有吸引力。上述预先部署、与运行时所需要的虚拟系统配置打包在一起的即运行应用称为虚拟应用。为了扩大此概念的适用范围，非常重要的一点是行业采用一种与供货商无关的标准，将上述虚拟设备和需要的元数据打包在一起，以便能够在任何虚拟平台上对其进行自动、安全的安装、配置和运行。

从用户的角度看，OVF 是一种虚拟应用的打包格式。在安装完成之后，OVF 包能够在用户的基础设施上添加一个自我包含、自我一致的软件应用，能够提供一项或多项具体功能。例如，OVF 包中可以包含有经过测试的完整功能的网页服务器、数据库和操作系统的组合，例如 LAMP 堆栈（Linux+Apache+MySQL+PHP），或者其中可以包含有病毒检测程序，包括其软件更新和间谍软件检测程序等。

尽管很多虚拟应用中都只包含有一套虚拟系统，但现代企业应用程序都采用服务导向架构（SOA）模型，其中包括多个层级，每个层级中都包含有一套或多套虚拟系统。因此，单一的虚拟系统模型不足以配置多层服务。除此之外，复杂的应用程序通常需要对网络安装时间进行自定义，同时还需要包括其他一些用户自定义功能。另外，使用运行时格式将虚拟应用与磁盘映像和配置数据打包在一起可能适合于特定的管理程序。运行时格式针对执行进行了优化，但未针对配置进行优化。因此，为了确保高效的软件配置，在其中加入一些额外功能是非常重要的，包括可移植性、平台独立性、验证、签署、版本管理、以及授权条款等。

OVF 规范提供了一种与管理程序无关的高效、可扩展和开放式的格式，能够用于打包和配置包含一套或多套虚拟系统的虚拟应用。其目标不仅在于促进单独的虚拟系统的自动、安全管理，也在于促进作为功能单元的虚拟应用的自动安全管理。

为了确保成功，OVF 获得了独立软件供货商、虚拟应用供货商、操作系统供货商、以及虚拟平台供货商的开发和支持。通过对不同供货商虚拟平台间的虚拟系统可移植性和可互通性常用标准的共同开发，OVF 规范增强了客户信心。



OVF 的主要目的在于提供立即生效、能够解决行业即时需求，促进一种通用、向后兼容的虚拟应用的打包格式的快速实施。

OVF 规范是与现有的 IT 管理标准和框架配套的规范，能够通过开放性和可扩展性促进最佳组合之间的竞争。本文件开头提供的版权通知的主要目的在于：在保证允许自由配置和实施本规范文件的同时，避免对 OVF 格式进行武断、独立、碎片化的扩展操作。

## 2. OVF 核心概念

### 2.1 虚拟应用

虚拟应用是指包含有一套或多套虚拟系统，经过预先配置的软件堆栈。每套虚拟系统都是一个可以独立安装的运行时实体，其中包括有操作系统、应用程序、以及应用程序的具体数据，还有虚拟系统要求用于描述相关虚拟硬件的元数据。很多可以通过网络进行访问的基础设施应用程序、甚至很多最终用户应用程序，例如 DNS 服务器，错误跟踪数据库，或者包括网页、应用和数据库层面的完整 CRM 解决方案，都可以采用虚拟应用形式交付。采用预先配置的软件堆栈的形式交付复杂的软件系统和服务，可以大大提高稳固性、降低安装过程的复杂性。

虚拟应用正在改变软件发布的范式，因为虚拟应用可以允许对具体应用程序的软件堆栈进行优化，向最终用户提供交钥匙服务。对于方案提供商而言，与构建硬件应用相比，构建虚拟应用更加简单、更具有成本效率。虚拟应用将软件和其使用的操作系统提前打包在一起，从而减少兼容性测试和认证，让软件能够预先安装在其运行的环境中（由开发相关解决方案的独立软件供货商提前安装）。对于最终用户而言，虚拟应用能够提供一个大程度简化软件生命周期管理的机会，具体方法为使用标准化、自动和高效的流程来替换之前使用的单个操作系统和应用的特定管理任务。

虚拟应用不一定需要由第三方独立软件供货商开发和交付这个概念同样非常重要。因为对于部分企业而言，企业内部可能存在由 IT 机构组装、测试和认证的针对具体服务的虚拟系统模板，之后即可以在企业内部按照“饼干模子”的方式进行重复打包。

通常情况下，软件服务是一个使用 SOA 模型在多套虚拟系统中运行、通过网络进行通信的多层级应用程序。软件服务中可能还包括其他服务，这些服务本身又是多层级的应用程序，或者其中又包括其他服务。实际上，SOA 模型与虚拟应用基础设施之间存在天然一致性，因为虚拟应用通常都会使用网络服务、XML 基管理和服务界面，这就可以将应用组合起来形成完整的应用程序。

例如，典型的网页应用程序中通常包括下列三个层面：一个实施表达逻辑的网页层面；一个实施业务逻辑的应用服务器层面；以及一个终端数据块库层面。最直接的实施方式就是将这些配置分解为三套虚拟系统，每个层面一套。通过这种方式，应用程序的规模将会从一台物理主机放大到三台物理主机上。另一种方法是将每个层面都视为服务本身。因此，每个层面都可以放大为多虚拟系统服务，从而提供集群式解决方案。这里同样再以网页应用程序为例，通常情况下网页应用程序包括很多网页服务器、少量的应用程序服务器、一到两台数据库服务器。在以虚拟系统的方式实施之后，每个层面上都可以根据要求设置或多或少的物理主机，每个层面都可以支持服务要求的多套虚拟系统实例。

一个 OVF 中可以包括一套或多套虚拟系统。具体哪种配置方式能够最契合其应用程序由开发商确定。OVF 可以在运行之前进行安装，虚拟平台可以通过 OVF 运行虚拟系统，但不要求使用此配置。如果选中了此配置，则 OVF 本身将不能再视为应用程序的“黄金镜像版”，因为虚拟系统的运行时状态将会渗透到 OVF 中。另外，此时允许虚拟平台检查 OVF 完整性的数字签名将无效。

作为传输机制，OVF 不同于威睿公司的 VMDK 虚拟磁盘格式和微软公司的 VHD 虚拟硬盘格式和开源的 QCOW 格式。上面所说的这些格式都是运行时虚拟系统镜像格式，只能在单独的虚拟硬盘上运行。尽管这些格式当前被频繁用作传输格式，但其本身并不是为了解决便携问题而设计的。这些格式不适用于具有多个硬盘的虚拟系统，也不适用于多套虚拟系统，在安装过程中还需要对虚拟系统进行自定义。如果虚拟系统需要在多个虚拟平台上运行，则也无法使用这些格式（即使虚拟平台声称支持所使用的具体虚拟硬盘格式）。

在 OVF 的恢复过程中，核心概念是所打包的虚拟应用的认证和完整性。此概念允许平台检测虚拟引用的来源，允许最终用户做出适宜的信任决策。OVF 规范中做出了相关规定，确保虚拟应用要对其自身的配置和修改负责。因此，这就意味着虚拟平台不需要能够读取虚拟应用的文件系统。虚拟平台和虚拟应用之间的解耦意味着可以使用任一虚拟系统来执行 OVF 包，也可以将其安装在任何支持 OVF 格式的虚拟平台上。在格式中提供了一个具体的机制，供虚拟应用检测其安装的平台并与平台互动。此机制允许虚拟平台在不破坏行业内部应用兼容性的情况下，扩展本规范的内容。

OVF 格式具有多种针对复杂、多层面服务，以及相关分配、安装、配置和执行的具体功能：

- 为多层面应用程序的配置、以及虚拟系统的组合提供直接支持，以便交付组合服务；
- 许可虚拟系统和应用程序层面配置规范；
- 具有稳固的 OVF 内容验证机制，完全支持无人操作安装，降低用户部署负担，进而提高用户体验；
- 使用商业标准流程通过签名和受信任的第三方对 OVF 内容进行完整性检查。此功能可以向用户保证自应用开发者签署之后，应用没有经过修改。此保证对于虚拟应用取得市场成功而言非常重要，对于虚拟应用的独立开发和在线下载而言也非常重要；
- 通过提供与应用相关的表达和确认授权条款的基本方法，尊重应用供货商和用户的商业利益。

## 2.2 寿命周期

虚拟系统的寿命周期如图 1 所示：

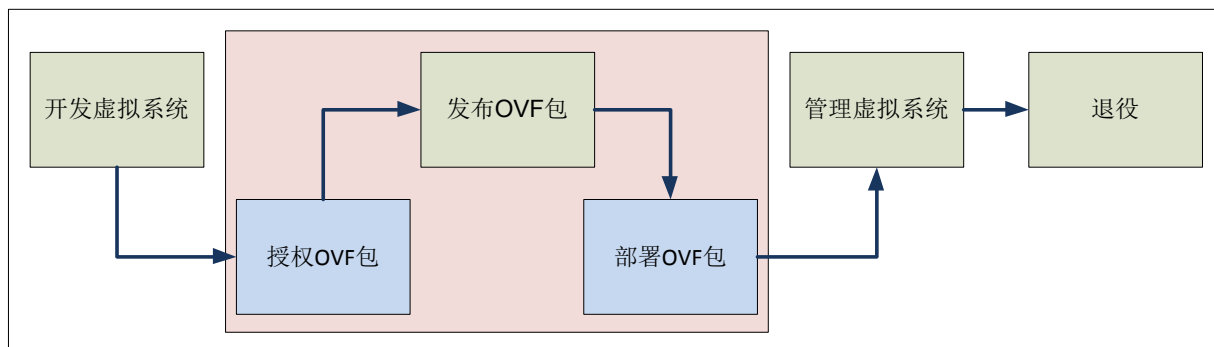


图 1: OVF 包生命周期

OVF 包是使用 OVF 作者开发或者购买的部件构建而成的。这些部件可以打包为一组构成虚拟应用的文件，其中包括一台或多台虚拟机和虚拟机集合，还包括相关的配置和部署元数据。例如，集群数据库部件可以从第三方独立软件供货商处购买。之后对安装的服务进行管理，直到其退役为止。发布、管理和退役不在本 OVF 规范文件范围之内，且与具体的虚拟产品和通过 OVF 安装的虚拟应用相关。管理包括对虚拟应用进行持续维护、配置和更新。具体工作内容取决于安装的服务和环境，与 OVF 包无关。OVF 规范文件重点关注授权和部署阶段。

OVF 的授权功能如图 2 相关内容所示：

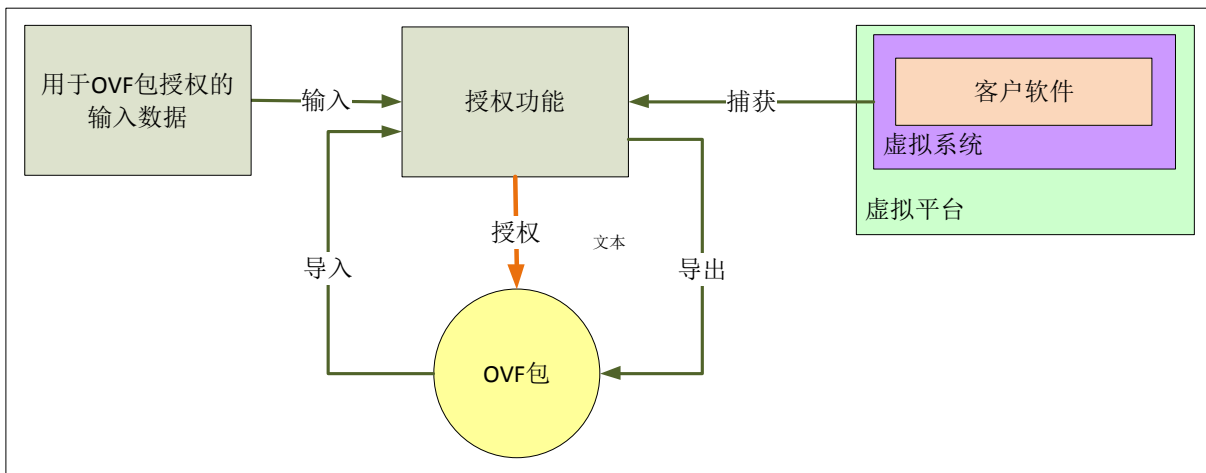


图 2: OVF 授权功能

OVF 包可以通过下列两种方式中的一种进行授权。最直接的方法是使用文本编辑器或者 XML 授权工具创建一个 OVF 描述文件，之后将其与需要的磁盘镜像和其他文件打包在一起，之后创建一个包含有 OVF 包的 tar 文件或者文件系统。

另外一种替代方法是从虚拟平台中导出 OVF 包。之后可以对 OVF 描述文件进行修改，纳入额外的信息。选择此方法可以出于多种原因，包括提高虚拟平台间的可移植性，或者提供配置方案。

OVF 的部署功能如图 3 相关内容所示。在此图中还说明了 OVF 工作小组的具体工作范围。

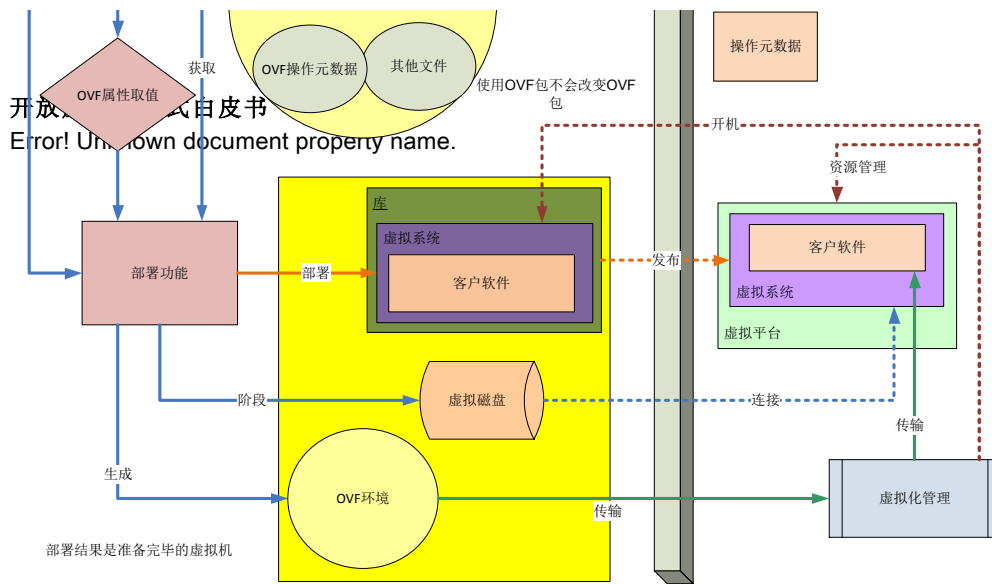


图 3： OVF 部署功能

OVF 操作元数据是确保虚拟系统或者虚拟系统集合正常运行所需要的信息。OVF 操作元数据是在虚拟系统启动之后，可用的操作元数据中的一个子集。

从上图中可以看出，通过 OVF 部署功能，可以将 OVF 环境传输到虚拟平台中。OVF 规范对于这种传输的性质没有做出硬性规定，但可以通过在虚拟机的虚拟读取器中放置媒体，例如 CD ROM 来实现，在每次启动之后，客户端操作系统都需要读取一遍媒体内容。OVF 环境中的元数据用于在操作系统启动之后进行必要配置，同时用于满足客户软件和虚拟平台的一些其他具体要求，以保证虚拟应用具有正常功能。

### 3. XML 后台

#### 3.1 XML 架构的使用

OVF 标准使用 XML 和 XML 架构定义语言 (XSD 或者 XSDL)。XML 是一种标记语言，对表达 XML 元素和属性的语法做出了规定，但对于文件中这些元素和属性的结构、或者元素和属性取值的允许数据类型没有做出多少强制性规定。文件类型定义 (DTD) 从一开始就是 XML 的一部分，主要目的是为 XML 架构添加结构和数据类型，但在很多情况下，这种定义并不充分。作为对上述情况的应对，W3C 开发了使用 XML 编制的 XSD，这是一种更为丰富的语言，对 XML 文件结构和数据类型做出了规定。XSD 和 DTD 可以同时使用，但 XSD 更加常用。XSD 文件通常具有 “.xsd” 后缀。

XSD 文件是描述 XML 文件中允许使用的元素和数据结构的元数据，XSD 还支持继承和其他对象导向结构。

Error! Unknown document property name.

## 开放虚拟化格式白皮书

这些结构能够简化详细、复杂的 XSD 文件，使其更容易编制、更紧凑、错误更少。根据 XML 语法检查 XML 文件只能测试文件的语法情况，而不能检查文件中的结构和数据的正确性。根据 XSD 能够检查文件中的结构和数据。在其他流程调用 XML 文件的情况下，根据 XSD 检查文件的有效性能确保文件具有更少的错误。除此之外，部分工具可以通过 XSD 生成 XML 文件。这些工具能够大大加快开发的速度。

OVF 使用 XSD 来规定 OVF 描述文件中的结构和数据类型，OVF 包中的 XML 文件用于描述虚拟系统安装的复杂流程。OVF 描述文件的后缀名为 “.ovf”。OVF 2.0.0 中使用的正式 XSD 的网址为：<http://schemas.dmtf.org/ovf/envelope/2/dsp8023.xsd>。OVF 用户应该使用 XSD 验证工具，根据 OVF XSD 架构来检查其 OVF 描述文件。描述文件成功通过验证是满足 OVF 标准的必要而非充分条件。在标准规范中还包括一些 dsp8023.xsd 之外的要求，因此必须手动检查才能够确保完全的合规性。

## 3.2 在 OVF 中使用的基本 XML 概念

### 3.2.1 元素

XML 元素是 OVF 描述文件中的数据容器。每个元素都不能为空，应该以一个起始标签开始，后接元素的内容，之后使用结束标签中止。起始标签中应该包括元素的名称，可以选择后接元素的属性，这些数值都需要放在尖括号内，例如<元素名称>或者<元素属性>。结束标签中只需要包括元素的名称，前面带一个斜杠/，放入尖括号内，例如</元素名称>。元素中还可以包括其他元素、文本内容、属性、或者所有上述内容的组合。

XML 元素应该遵守下列命名规则：

- 名称中可以带有字符、数字、以及其他符号；
- 名称不能以数字或者标点符号开始；
- 名称不能以 xml（或者 XML、或者 Xml 等）字符开始；
- 名称中不得带有空格；
- 任何名称都可以使用，不保留任何字符。

### 3.2.2 属性

XML 元素中可以带有属性，以便提供关于元素的其他额外信息。属性通常能够提供不是数据一部分的相关信息。属性取值应该随时放入到单引号或者双引号中。

### 3.2.3 置换群

置换群是 XSD 中的一个对象导向功能，允许你对可以使用通过架构生成的文件中的其他元素进行置换的元素。可置换元素被称为数据头元素，应该在架构的范围内做出规定。置换群内的元素应该具有与数据头元素相同的类型，或者通过数据头类型衍生出来的类型。

从本质上而言，置换群可以让你构建一组可以替换通用元素的元素集合。例如，如果你想要构建一套销售

Error! Unknown document property name.

三种类型小饰品的公司订货系统，则你可以定义一个通用小饰品元素，其中包括所有三种小饰品类型通用的数据。之后，你可以定义一个通用小饰品元素的置换群，其中包含可以从通用元素衍生出来的更加具体的信息。在架构中，可以将订单定义为一系列的通用饰品元素的组合。在遵守架构的 XML 文件中，订单可以采用置换群而非通用元素中的相关元素。通常情况下，置换群的数据头元素构成摘要，因此通用属性元素不会在 XML 文件中出现。

## 4. OVF 包

OVF 包提供了一种在虚拟系统或者虚拟系统集合中发布软件解决方案的途径。OVF 包中包括有 OVF 描述文件和相关的虚拟磁盘。OVF 包可以为通过 URL 引用的一组文件，也可以为带有“.ova”后缀名的压缩文件。

### 4.1 OVF 描述文件结构

OVF 描述文件是一个 XML 文件。OVF 描述文件的根元素是 Envelope（包络）。Envelope 元素中最重要的两个子元素是 VirtualSystem（虚拟系统）元素和 VirtualSystemCollection（虚拟系统集合）元素。Envelope 元素中还包含有适用于 VirtualSystem 和 VirtualSystemCollection 元素的字段。Envelope 元素中可以同时包含 VirtualSystem 和 VirtualSystemCollection 元素。VirtualSystemCollection 元素具有递归性结构，因此和 Envelope 元素一样，其中也可以同时包含 VirtualSystem 和 VirtualSystemCollection 元素。OVF 架构中规定了各种不同类型的字段。其中部分字段只能出现在 Envelope 元素中。其他字段可以出现在 VirtualSystem 元素中；其他字段只能出现在 VirtualSystemCollection 元素中。部分字段可以同时出现在 VirtualSystem 和 VirtualSystemCollection 元素中。具体的结构如图 4 相关内容所示。

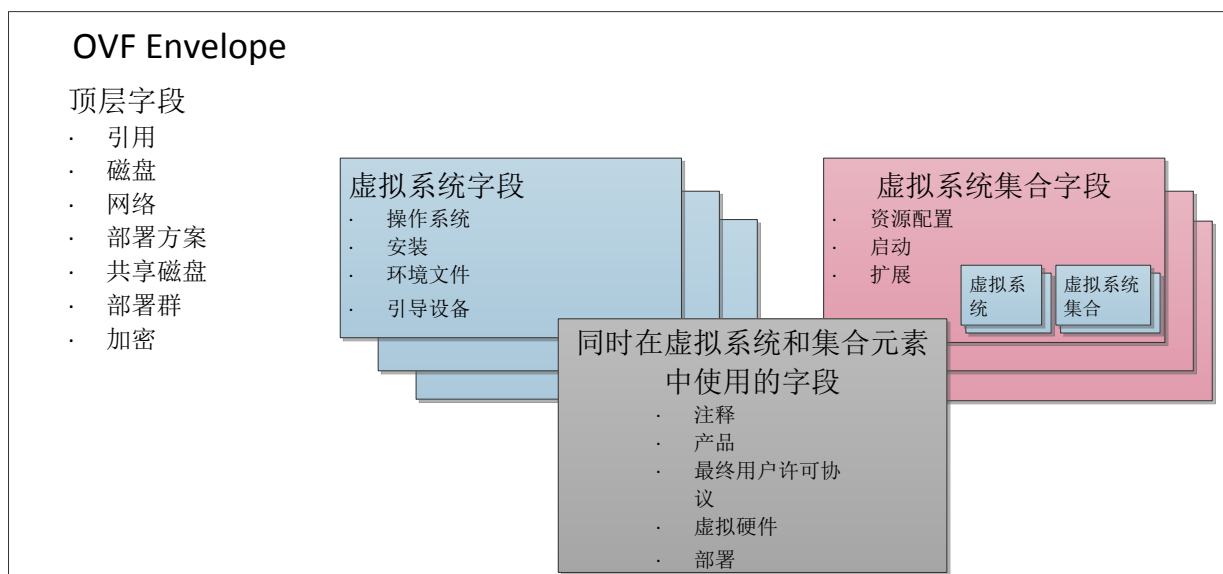


图 4: OVF 包结构

## 4.2 OVF 架构中规定的全局属性

在 OVF 架构中定义的下列属性为全局属性，另外还对具体 OVF 属性的 OVF 元素进行了定义（参阅第 4.4 节和第 4.3 节相关内容）：

- **Required（要求）属性：**表示如果没有此元素、或者此元素无法理解，则部署将会失败。此属性是一个 XSD 布尔值，允许取值为“true”、“false”、“0”、“1”。如果未特别说明，则其取值为“true”或者“1”。此 OVF 属性不得与 XSD 的用途（Use）属性的 Required（要求）取值向混淆。这两个术语虽然相似，但重要性完全不同。
- **Transport（传输）属性：**这是一个使用空格分隔的、支持的传输类型的列表，用于向客户端软件提供信息。更多详细信息参阅第 4.7.3 节和第 6.2 节相关内容。
- **Configuration（配置）属性：**用于识别在 DeploymentOptionSection（部署方案字段）元素中的 Configuration（配置）元素中的配置，参阅第 4.4.6 节相关内容。Configuration（配置）属性可以用在下面这些地方：
  - ◆ 用于 Item（项目）、EthernetPortItem（以太网端口项）、StorageItem（存储器项）元素中的 VirtualHardwareSection（虚拟硬件字段）元素中；
  - ◆ 用于 Item（项目）、EthernetPortItem（以太网端口项）、StorageItem（存储器项）元素中的 ResourceAllocationSection（资源配置字段）元素中；
  - ◆ 用于 InstanceCount（实例计数）元素中的 ScaleOutSection（扩展字段）元素中；
  - ◆ 用于 Property（属性）元素中的 ProductSection（产品字段）元素中。
- **Bound（范围）属性：**是一个范围标记属性，用于显示设定数据资源配置的最大值、正常值和最小值。其允许取值为“min（最小值）”、“normal（最大值）”和“max（最大值）”，参阅第 4.6.1 节相关内容。Bound（范围）属性可以用于下面两个地方，用于设定资源配置的限值：
  - ◆ 用于 Item（项目）、EthernetPortItem（以太网端口项）、StorageItem（存储器项）元素中的 VirtualHardwareSection（虚拟硬件字段）元素中；
  - ◆ 用于 Item（项目）、EthernetPortItem（以太网端口项）、StorageItem（存储器项）元素中的 ResourceAllocationSection（资源配置字段）元素中。

## 4.3 OVF 可扩展性

OVF 架构使用 XSD 中的“any（任何）”和“anyAttribute（任何属性）”来扩展 OVF 描述文件和 OVF 环境，以便提供自定义元数据。此功能将允许 OVF 包满足行业中各种用途实例的具体需求。

下列定义来自于 XML 架构参考网站，网址为：

[http://www.w3schools.com/schema/schema\\_elements\\_ref.asp](http://www.w3schools.com/schema/schema_elements_ref.asp)。

- Any（任何）：此定义用于帮助作者使用架构中没有规定的元素来扩展 XML 文件；
- anyAttribute（任何属性）：此定义能够帮助作者使用架构中没有规定的属性来扩展 XML 文件；
- ##any：允许使用来自任何命名空间的元素（此定义为默认定义）；
- ##other：允许使用除母元素命名空间之外的任何命名空间中的元素。

在 Envelope（包络）元素层面上进行扩展时，可以通过在置换群中定义一个新的 ovf 字段成员来实现。在 Content（内容）元素层面上进行扩展时，可以通过在置换群中定义一个新的 ovf 字段成员来实现。参阅第 3.2.3 节中相关内容。在 OVF 架构允许出现这些字段的情况下，可以使用这些新的字段元素。每个新的字段元素中的 Info（信息）元素可以用于在部署平台不能理解新的字段元素，从而将其跳过的情况下，为客户提供警告信息。

可以在尾部加入其他元素，来对 OVF 架构规定的元素进行扩展。扩展点使用带有 namespace="##other" 的 xs:any 进行宣告。

在 OVF 架构中还允许使用其他属性，使用 xs:anyAttribute 宣告扩展点。

ovf:required（要求）属性用于规定元素中的这些信息是强制要求的还是可选的。ovf:required 属性的默认取值为 TRUE。如果部署平台检测到其不能理解的强制要求的元素扩展，则部署将会失败。

对于自定义属性而言，属性中的信息并不强制要求提供，以确保系统具有正确的行为。

EXAMPLE 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>
```

EXAMPLE 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>
```



Error! Unknown document property name.

## 开放虚拟化格式白皮书

EXAMPLE 3:

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

### 4.4 OVF 顶层元素

OVF 架构中定义的根元素为 **Envelop**（包络）元素。在 OVF 描述文件中出现、属于 **Envelop** 元素直接子元素的 OVF 元素按照下面的顺序列出：

- 引用元素；
- 字段元素：一个字段元素的置换群；
- 内容元素：一个内容元素的置换群；
- 字符串元素。

构成 **Section**（字段）元素置换群的元素是 **Envelope** 元素的直接子元素：

- **DiskSection**（磁盘字段）元素；
- **NetworkSection**（网络字段）元素；
- **DeploymentOptionSection**（部署选项字段）元素；
- **SharedDiskSection**（共享磁盘字段）元素；
- **PlacementGroupSection**（布置组字段）元素；
- **EncryptionSection**（加密字段）元素。

构成 **Content**（内容）元素置换群的元素包括：

- **VirtualSystem**（虚拟系统）元素；
- **VirtualSystemCollection**（虚拟系统集合）元素。

构成在 **Content**（内容）元素中使用的 **Section**（字段）元素置换群的元素，按照其在 OVF 描述文件中出现的顺序在下面列出：

- **AnnotationSection**（注释字段）元素；
- **ProductSection**（产品字段）元素；
- **OperatingSystemSection**（操作系统字段）元素；
- **EulaSection**（最终用户许可协议字段）元素；

Error! Unknown document property name.

- VirtualHardwareSection（虚拟硬件字段）元素；
- ResourceAllocationSection（资源集合字段）元素；
- InstallSection（安装字段）元素；
- StartupSection（启动字段）元素；
- EnvironmentFilesSection（环境文件字段）元素；
- BootDeviceSection（引导设备字段）元素；
- ScaleOutSection（扩展字段）元素；
- PlacementSection（布置字段）元素。

在 OVF 架构中规定的其他元素如下所述。需要注意的一点是，这种元素可以在除 OVF 之外的命名空间内使用：

- Annotation（注释）；
- AppUrl；
- bootc:CIM\_BootConfigSetting（引导配置设置）；
- Category（分类）；
- Configuration（配置）；
- Content（内容）；
- Description（描述）；
- Disk（磁盘）；
- EthernetPortItem（以太网端口）；
- File（文件）；
- FullVersion（完整版本）；
- Icon（图标）；
- Info（信息）；
- InstanceCount（实例计数）；
- Item（项目）；
- Label（标签）；

Error! Unknown document property name.

## 开放虚拟化格式白皮书

- License (许可);
- Msg (消息);
- Name (姓名);
- Network (网络);
- NetworkPortProfile (网络端口概况);
- NetworkPortProfileURI (网络端口概况 URI);
- Product (产品);
- ProductUrl (产品 Url);
- Property (属性);
- SharedDisk (共享磁盘);
- StorageItem (存储项);
- System (系统);
- Value (取值);
- Vendor (供货商);
- VendorUrl (供货商 Url);
- Version (版本);
- xenc:EncryptedKey (加密密钥);
- xenc11:DerivedKey (衍生密钥)。

图 4 中所示的 OVF 描述文件的基本结构示例如下所述:

```
ovf:Envelope
<xs:element name="References" type="ovf:References_Type">
<xs:element ref="ovf:Section" minOccurs="0" maxOccurs="unbounded">
<DiskSection>
  <Info> Describes all virtual disks used with the package </Info>
<NetworkSection>
  <Info>List of logical networks used in the package</Info>
<DeploymentOptionSection>
  <Info>List of deployment options available in the package</Info>
```

```

<xs:element ref="ovf:Content">
  <VirtualSystemCollection ovf:id="Acme VSC">
    <Info>The packaging of the first virtual appliance</Info>
    -----
    <VirtualSystem ovf:id="Acme VS 1">
      <Info>The packaging of the virtual machine 1</Info>
    <VirtualSystem ovf:id="Acme VS 2">
      <Info>The packaging of the virtual machine 2</Info>
    -----
    <VirtualSystemCollection ovf:id="Widget VSC">
      <Info>The packaging of the second virtual appliance</Info>
      <VirtualSystem ovf:id="Acme VS 3">
        <Info>The packaging of the virtual machine 3</Info>
      <VirtualSystem ovf:id="Acme VS 3">
        <Info>The packaging of the virtual machine 4</Info>
      -----
    <xs:element name="Strings" type="ovf:Strings_Type" minOccurs="0" maxOccurs="unbounded">
      <Info> Root element of I18N string bundle</Info>
  
```

#### 4.4.1 VirtualSystem 元素

VirtualSystem（虚拟系统）元素是 Content（内容）元素的一个置换元素。其中含有一定数量的 Section（字段）元素，用于定义单套虚拟系统。这些字段元素将用于描述适用于虚拟系统的虚拟硬件、资源配置、以及产品信息。

#### 4.4.2 VirtualSystemCollection 元素

VirtualSystemCollection（虚拟系统集合）元素是 Content（内容）元素的一个置换元素。其中包括有一个或者多个 VirtualSystem（虚拟系统）元素，以及一定数量的 Section（字段）元素，用于定义一个虚拟系统集合。这些字段元素将用于描述适用于虚拟系统集合的虚拟硬件、资源配置、以及产品信息。

#### 4.4.3 References 元素

References（引用）元素中含有对所有外部文件的引用信息。

#### 4.4.4 DiskSection 元素

DiskSection（磁盘字段）元素用于在 OVF 包中，定义虚拟系统使用的虚拟磁盘。

可以使用任何虚拟磁盘，只要相关虚拟磁盘格式规范公开、可以不受限制地使用。此元素支持当前管理程序使用的所有虚拟硬盘格式，而且在未来可以扩展到其他格式。

Error! Unknown document property name.

## 开放虚拟化格式白皮书

虚拟磁盘格式可以为所安装的客户端软件不了解的简单、基本磁盘模块格式。例如，威睿公司的 VMDK 格式能够处理 512 字节磁盘区域，储存在 64KB 模块中的一些数量的稀疏、压缩变量中。在部署之时，虚拟平台能够使用其优选的基本磁盘模块格式创建虚拟磁盘。运行时虚拟磁盘格式可以与发布格式相同，但通常都会有所不同，因为无法高效地运行压缩后的虚拟磁盘格式。客户端软件中也可以安装自己的文件系统格式，例如 NTFS、EXT3、或者 ZFS。OVF 虚拟磁盘不需要知道上述文件系统格式。

下列示例中展示了虚拟磁盘描述：

```
<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>
```

### 4.4.5 NetworkSection 元素

NetworkSection（网络字段）元素中描述了在部署 OVF 包时需要使用的网络和网络连接。网络可以通过简单化的术语进行定义，例如红绿蓝网络。以太网端口特性通过 CIM\_EthernetPortAllocationSettingData（CIM\_以太网端口配置设置数据）分级进行定义。在其中可以使用“网络端口概况（DSP 8049）”进行更加详细的定义，上述规范也可以包括在 OVF 包中。

此元素有一个基本假设就是：存在可以通过以太网端口进行连接的平层 2 网络。但不对该网络的服务或者特性做出任何假设。以太网端口的速度可以使用 CIM\_EthernetPortAllocationSettingData（CIM\_以太网端口配置设置数据）进行设置。

#### 4.4.5.1 OVF 网络

下面这个示例是网络字段的 OVF 描述文件中的一个判断。这个示例中简单地说明了虚拟系统的以太网端口连接到“红色”网络上。在 epasd:Connection 属性中，规定了与以太网端口连接的具体网络。这里就简单地假设所有以太网端口都连接到平层 2 网络上。如果规定了多个网络，例如“红色”和“绿色”网络，则需要假设“红色”和“绿色”网络没有被连接。与外部网络的连接（例如连接数据中心 LAN 或者 WAN）被认为属于部署阶段的一部分，因此没有在 OVF 描述文件中进行规定。在这种情况下，OVF 包的用户应该与主机服务器达成协议，确定网络的具体特性和服务。

Error! Unknown document property name.

```
<!-- Describes all networks used in the package -->  
<NetworkSection>  
  <Info>Logical networks used in the package</Info>  
  <Network ovf:name="Red Network">  
    <Description>The network that the virtual systems are attached to.  
    on</Description>  
  </Network>  
</NetworkSection>
```

下面这个示例是 OVF 描述文件中 VirtualSystem（虚拟系统）元素中的 VirtualHardwareSection（虚拟硬件字段）中的一个项目的片段，其中解释了如何使用 CIM\_EthernetPortAllocationSettingData（CIM\_以太网端口配置设置数据）分级中的 Connection（连接）属性，将虚拟系统连接到“红色”网络上。

```
<EthernetPortItem>  
  <epasd:AddressOnParent>7</epasd:AddressOnParent>  
  <epasd:AutomaticAllocation>true</epasd:AutomaticAllocation>  
  <epasd:Connection>Red Network</epasd:Connection>  
  <epasd:Description>Virtual Ethernet adapter</epasd:Description>  
  <epasd:ElementName>Virtual NIC 1</epasd:ElementName>  
  <epasd:InstanceID>8</epasd:InstanceID>  
  <epasd:ResourceType>10</epasd:ResourceType>  
</EthernetPortItem>
```

可以定义多个网络。

图 5 中列出了一个简单化网络，其中以太网端口与“红色”网络连接。

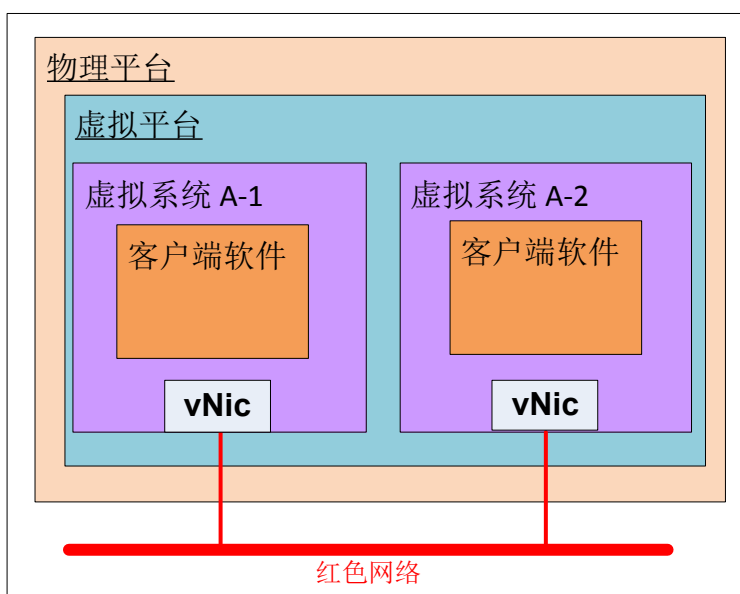


图 5：网络连接

图 6 中显示了双重网络配置的情况：其中“绿色”网络用于存储连接，“红色”网络用于局域网连接。

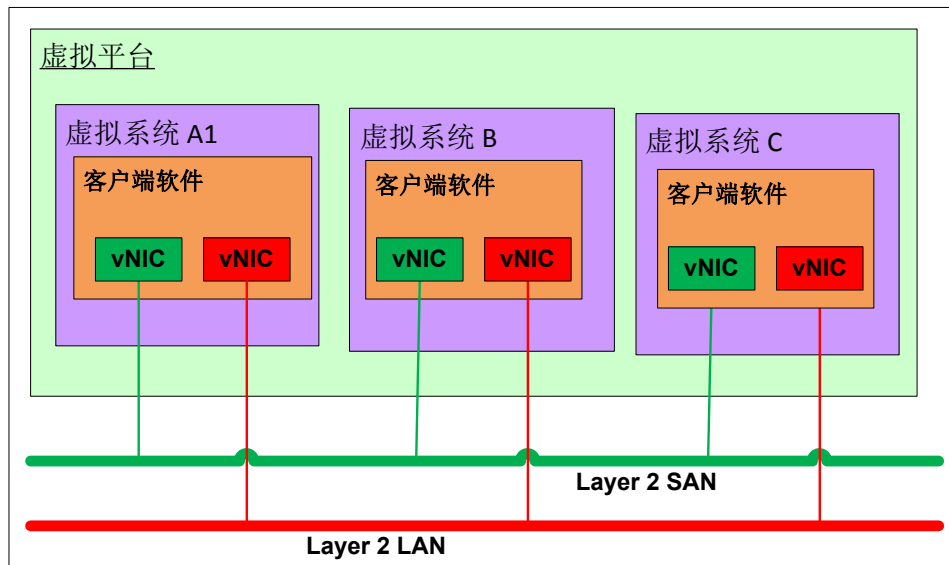


图 6：LAN-SAN 网络连接

#### 4.4.5.2 网络端口概况

在“网络端口概况（DSP 8049）”中，对 OVF 包中定义的虚拟系统通信时使用的网络端口属性配置做出的规定。更多相关信息参阅“虚拟网络管理白皮书（DSP 2025）”相关内容。

在 EthernetPortAllocationSettingData（以太网端口配置设置数据）中的完整属性清单如下所述。每个属性的允许取值都将在 DSP 0243、DSP 1041、DSP 1050、以及 CIM 架构中做出定义。在绝大多数 OVF 描述文件中，只会使用其中一部分的属性。

```
<EthernetPortItem>
  <epasd:Address>
  <epasd:AddressOnParent>
  <epasd:AllocationUnits>
  <epasd:AllowedPriorities>
  <epasd:AllowedToReceiveMACAddresses>
  <epasd:AllowedToReceiveVLANs>
  <epasd:AllowedToTransmitMACAddresses>
  <epasd:AllowedToTransmitVLANs>
  <epasd:AutomaticAllocation>
  <epasd:AutomaticDeallocation>
  <epasd:Caption>
  <epasd:ChangeableType>
  <epasd:ConfigurationName>
```

```

<epasd:Connection>
<epasd:ConsumerVisibility>
<epasd:DefaultPortVID>
<epasd:DefaultPriority>
<epasd:Description>
<epasd:DesiredVLANEndpointMode>
<epasd:ElementName>
<epasd:GroupID>
<epasd:HostResource>
<epasd:InstanceID>
<epasd:Limit>
<epasd:ManagerID>
<epasd:MappingBehavior>
<epasd:NetworkPortProfileID>
<epasd:NetworkPortProfileIDType>
<epasd:OtherEndpointMode>
<epasd:OtherNetworkPortProfileIDTypeInfo>
<epasd:OtherResourceType>
<epasd:Parent>
<epasd:PoolID>
<epasd:PortCorrelationID>
<epasd:PortVID>
<epasd:Promiscuous>
<epasd:ReceiveBandwidthLimit>
<epasd:ReceiveBandwidthReservation>
<epasd:Reservation>
<epasd:ResourceSubType>
<epasd:ResourceType>
<epasd:SourceMACFilteringEnabled>
<epasd:VSITypeID>
<epasd:VSITypeIDVersion>
<epasd:VirtualQuantity>
<epasd:VirtualQuantityUnits>
<epasd:Weight>
</EthernetPortItem>

```

#### 4.4.6 DeploymentOptionsSection 元素

DeploymentOptionsSection（部署选项字段）元素是 Envelope（包络）元素的直接子元素。而 Configuration



Error! Unknown document property name.

## 开放虚拟化格式白皮书

(配置)元素又是 `DeploymentOptionsSection`(部署选项字段)元素的直接子元素。`DeploymentOptionsSection`(部署选项字段)元素中含有一个或多个 `Configuration`(配置)元素。

`Configuration`(配置)元素用于在部署 OVF 包时,规定所使用的资源配置。可以在部署之时选择一种配置。在使用 `userConfigurable`(用户可配置)属性时,可以要求用户提供输入信息。

`DeploymentOptionsSection`(部署选项字段)元素中列出了 OVF 包中可用的配置的 ID、标签和描述信息。

通过将 `default`(默认)属性的取值设置为 `true`,可以启用默认配置。在没有其他输入信息的情况下,系统将使用默认配置。如果没有启用默认配置,则系统将会把第一项配置视为默认配置。

每项配置都具有一个独特的 ID 属性,用于识别该配置。此 ID 属性的取值可以在其他字段元素的 `configuration`(配置)属性中做出定义,例如 `VirtualHardwareSection`(虚拟硬件字段)元素、`ProductSection`(产品字段)元素、以及 `ScaleOutSection`(扩展字段)元素。

如果某个元素具有多项配置,则此元素的 `configuration`(配置)属性的取值可以为使用空格分隔的多个配置 ID 列表。

部署功能需要从用户输入信息、或者从其他元数据中选取一个配置。在选中配置之后,具有 `configuration`(配置)属性的元素将被选择为部署时使用的配置。没有 `configuration`(配置)属性的元素也可以进行不是,但不需要特别为其提供一个不同的配置属性。因此,在部署过程中选择特定属性可能会对 OVF 包中的不同字段中的不同项目的配置造成影响。

下面是 OVF 描述文件中的 `DeploymentOptionsSection`(部署选项字段)元素的片段,其中解释了 `Configuration`(配置)元素的使用情况。需要注意的一点是,为默认配置分配的内存资源以蓝色标记,为“Big”配置分配的资源以灰色标记。除非用户通过输入信息选择“Big”配置,否则部署功能将选择默认配置。

```
<DeploymentOptionsSection>
  <Configuration ovf:id="big">
    <Label>Big</Label>
    <Description>Apply reservations for Memory</Description>
  </Configuration>
  ...
</DeploymentOptionsSection>

<VirtualHardwareSection>
  <Info>...</Info>

  <Item>
    <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
    <rasd:ElementName>512 MB memory size-no reservation</rasd:ElementName>
    <rasd:InstanceID>0</rasd:InstanceID>
    <rasd:ResourceType>4</rasd:ResourceType>
    <rasd:VirtualQuantity>512</rasd:VirtualQuantity>
  </Item>
```

Error! Unknown document property name.

```
<Item ovf:configuration="big">
  <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
  <rasd:ElementName>512 MB memory size & 256 MB reservation</rasd:ElementName>
  <rasd:InstanceID>0</rasd:InstanceID>
  <rasd:Reservation>256</rasd:Reservation>
</Item>
</VirtualHardwareSection>
```

最终得到的资源配置 CIM 实例如下所述:

```
<rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
<rasd:ElementName>512 MB memory size & 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>
```

其中使用灰色高亮标记的文本是“Big”配置，使用蓝色高亮标记的文本是默认配置。

下面这个示例中描述了带有三个 Configuration（配置）元素的 DeploymentOptionsSection（部署选项字段）元素。需要注意的一点是，“Normal”配置被设置为了默认配置，因此在没有配置被选中的情况下，部署功能使用了 Configuration（配置）属性取值为“normal（标准）”的元素。

```
<DeploymentOptionSection>
  <Configuration ovf:id="minimal">
    <Label>Minimal</Label>
    <Description>Smallest practical implementation</Description>
  </Configuration>
  <Configuration ovf:id="normal" ovf:default="true">
    <Label>Normal</Label>
    <Description>A typical implementation</Description>
  </Configuration>
  <Configuration ovf:id="large">
    <Label>Large</Label>
    <Description>A scaled up implementation</Description>
  </Configuration>
</DeploymentOptionSection>
```

在下面这个示例中，描述了使用 Configuration（配置）属性来定义上述三种配置的资源配置。其中 VirtualHardwareSection（虚拟硬件字段）元素使用 CIM\_ResourceAllocationDescriptor（CIM\_资源配置描述文件）属性来规定想要的内存资源配置。VirtualHardwareSection（虚拟硬件字段）中的每个 Item（项目）

Error! Unknown document property name.

## 开放虚拟化格式白皮书

元素都是指在上文中所述的 `DeploymentOptionsSection`（部署选项字段）中定义的一个配置。

```
<VirtualHardwareSection>
  <Info>...</Info>
  <Item ovf:configuration="normal">
    <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="minimal">
    <rasd:AllocationUnits>byte * 2^20</rasd:AllocationUnits>
    <rasd:ElementName>256 MB memory size and 128 MB reservation</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:Reservation>128</rasd:Reservation>
    <rasd:ResourceType>4</rasd:ResourceType>
    <rasd:VirtualQuantity>256</rasd:VirtualQuantity>
  </Item>
  <Item ovf:configuration="large">
    <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>
```

在上面的示例中，通过部署开始时选择的配置来控制内存尺寸。如果没有选择配置，则内存尺寸将为默认的“normal（标准）”配置。

下面这个示例说明了 `ProductSection`（产品字段）中可配置属性的用途。

```
<ProductSection>
  <Property ovf:key="app_log" ovf:type="string" ovf:value="low"
    ovf:configuration="normal">
    <Label>Loglevel</Label>
    <Description>Loglevel for the service</Description>
```

Error! Unknown document property name.

```
<Value ovf:value="none" ovf:configuration="minimal">
  <Value ovf:value="high" ovf:configuration="large">
</Property>
</ProductSection>
```

在此示例中，“app\_log”的取值会随着所选择的配置不同而变化。在使用虚拟硬件的情况下，如果没有选中配置，则“app\_log”属性的默认取值为“normal（标准）”配置取值，即“low”。

#### 4.4.7 SharedDiskSection 元素

SharedDiskSection（共享磁盘字段）元素允许多套虚拟系统引用虚拟磁盘，以便满足集群数据库的需求。所使用的具体文件共享技术可以由平台决定。

SharedDiskSection（共享磁盘字段）元素只有在 envelope（包络）层面上有效。

每个共享磁盘都有一个独特的 OVF 包识别符。SharedDiskSection（共享磁盘字段）元素中加入了一个布尔值 ovf:readOnly（只读）属性，用于表示该元素为可读写（取值为 FALSE）还是只读（取值为 TRUE）。

下面这个示例显示了 SharedDiskSection（共享磁盘字段）元素的基本情况：

```
<ovf:SharedDiskSection>
  <Info>Describes the set of virtual disks shared between VMs</Info>
  <ovf:SharedDisk ovf:diskId="datadisk" ovf:fileRef="data"
    ovf:capacity="8589934592" ovf:populatedSize="3549324972"
    ovf:format="http://www.vmware.com/interfaces/specifications/vmdk.html#sparse"/>
  <ovf:SharedDisk ovf:diskId="transientdisk" ovf:capacity="536870912"/>
</ovf:SharedDiskSection>
```

下面这个示例中解释了共享磁盘的使用。用于安装操作系统（system）、集群软件（crs\_home）和数据块（db\_home）的磁盘通过引用外部文件进行支持。此示例中的共享磁盘没有通过外部文件引用进行支持。部署功能将创建由多套虚拟系统共享的共享磁盘。

```
<ovf:References>
  <ovf:File ovf:id="system" ovf:href="system.img" ovf:compression="gzip"/>
  <ovf:File ovf:id="crs_home" ovf:href="crs_home.img" ovf:compression="gzip"/>
  <ovf:File ovf:id="db_home" ovf:href="db_home.img" ovf:compression="gzip"/>
</ovf:References>
<ovf:DiskSection>
  <ovf:Info>Virtual Disks</ovf:Info>
  <ovf:Disk ovf:diskId="system" ovf:fileRef="system" ovf:capacity="5368709120"
  ovf:format="Raw disk image"/>
  <ovf:Disk ovf:diskId="crs_home" ovf:fileRef="crs_home" ovf:capacity="2147483648"
  ovf:format="Raw disk image"/>
```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<ovf:Disk ovf:diskId="db_home" ovf:fileRef="db_home" ovf:capacity="4294967296"
ovf:format="Raw disk image"/>
</ovf:DiskSection>
<ovf:SharedDiskSection>
  <ovf:Info>Virtual Disks shared at runtime</ovf:Info>
  <ovf:SharedDisk ovf:diskId="crs_asm" ovf:capacity="4294967296" ovf:format="Raw disk
image"/>
  <ovf:SharedDisk ovf:diskId="db_asm" ovf:capacity="12884901888" ovf:format="Raw disk
image"/>
</ovf:SharedDiskSection>
.....
<ovf:VirtualSystemCollection ovf:id="rac_db_asm">
  <ovf:Info>Sample Oracle RAC using ASM</ovf:Info>
  .....
  <ovf:ScaleOutSection ovf:id="rac_db">
    <ovf:Info>RAC DB</ovf:Info>
    <ovf:Description>Number of instances</ovf:Description>
    <ovf:InstanceCount ovf:default="2" ovf:minimum="2" ovf:maximum="4"</ovf:InstanceCount>
  </ovf:ScaleOutSection>
  .....
  <ovf:VirtualSystem ovf:id="rac_db">
    <ovf:Info>RAC DB Instance</ovf:Info>
    .....
    <ovf:VirtualHardwareSection>
      <ovf:Info>System requirements: 8192 MB, 2 CPUs, 5 disks, 2 nics
      </ovf:Info>
      .....
      <ovf:Item>
        <rasd:Description>Disk 1</rasd:Description>
        <rasd:ElementName>Disk 1</rasd:ElementName>
        <rasd:HostResource>ovf:/disk/system</rasd:HostResource>
        <rasd:ResourceType>17</rasd:ResourceType>
      </ovf:Item>
      <ovf:Item>
        <rasd:Description>Disk 2</rasd:Description>
        <rasd:ElementName>Disk 2</rasd:ElementName>
        <rasd:HostResource>ovf:/disk/crs_home</rasd:HostResource>
        <rasd:ResourceType>17</rasd:ResourceType>
```

Error! Unknown document property name.

```

</ovf:Item>
<ovf:Item>
  <rasd:Description>Disk 3</rasd:Description>
  <rasd:ElementName>Disk 3</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/db_home</rasd:HostResource>
  <rasd:ResourceType>17</rasd:ResourceType>
</ovf:Item>
<ovf:Item>
  <rasd:Description>Disk 4</rasd:Description>
  <rasd:ElementName>Disk 4</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/crs_asm</rasd:HostResource>
  <rasd:ResourceType>17</rasd:ResourceType>
</ovf:Item>
<ovf:Item>
  <rasd:Description>Disk 5</rasd:Description>
  <rasd:ElementName>Disk 5</rasd:ElementName>
  <rasd:HostResource>ovf:/disk/db_asm</rasd:HostResource>
  <rasd:ResourceType>17</rasd:ResourceType>
</ovf:Item>
.....
</ovf:VirtualHardwareSection>
</ovf:VirtualSystem>
</ovf:VirtualSystemCollection>

```

#### 4.4.8 PlacementGroupSection 元素

PlacementGroupSection（布置组字段）元素中规定了布置组的 ID 及其相关的布置政策。通过使用 PlacementSection（布置字段）元素，可以在布置组与 VirtualSystemCollection（虚拟系统集合）或者 VirtualSystem（虚拟系统）之间建立相关关系。

PlacementGroupSection（布置组字段）元素的示例如下所述：

```

<ovf:PlacementGroupSection ovf:id="PG2" ovf:policy="availability">
  <Info>Placement policy for group of virtual systems that need availability</Info>
  <ovf:Description>Placement policy for a database tier</ovf:Description>
</ovf:PlacementGroupSection>
...
<ovf:PlacementGroupSection ovf:id="PG1" ovf:policy="affinity">
  <Info>Placement policy for group of virtual systems that need affinity</Info>
  <ovf:Description>Placement policy for a web tier</ovf:Description>

```

```
</ovf:PlacementGroupSection>
```

PlacementGroupSection（布置组字段）元素是 Envelope（包络）元素的直接子元素。参阅第 4.5.5 节相关内容。

## 4.5 在虚拟系统和虚拟系统集合中使用的 OVF 字段元素

在 VirtualSystem（虚拟系统）或者 VirtualSystemCollection（虚拟系统集合）中可以出现下列 OVF 描述文件字段元素。

### 4.5.1 AnnotationSection 元素

AnnotationSection（注释项字段）元素是一个用户自定义元素，可以出现在 VirtualSystem（虚拟系统）和 VirtualSystemCollection（虚拟系统集合）元素中。AnnotationSection（注释项）元素中包含有一个 Annotation（注释）元素。Annotation（注释）元素可以本地化。建议在部署之时，使用 Annotation（注释）元素向客户提供必要的信息。需要注意的一点是，OVF 中规定的 Annotation（注释）元素并不是 XML 架构的 Annotation（注释）元素。

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

### 4.5.2 ProductSection 元素

ProductSection（产品字段）元素中将会提供一些产品信息，例如相关虚拟应用的名称和供货商，以及一组可以用于对虚拟应用进行自定义的属性。这些属性可以在虚拟应用安装时进行配置，通常是由用户进行配置。这些内容将在后文中进一步进行详细讨论。

```
<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="adminEmail" ovf:type="string" ovf:userConfigurable="true">
    <Label>Admin email</Label>
    <Description>Email address of administrator</Description>
  </Property>
```

```
<Category>Admin properties</Category>
<Property ovf:key="appLog" ovf:type="string" ovf:value="low" ovf:userConfigurable="true">
  <Description>Loglevel for the service</Description>
</Property>
<Property ovf:key="appisSecondary" ovf:value="false" ovf:type="boolean">
  <Description>Cluster setup for application server</Description>
</Property>
<Property ovf:key="appIp" ovf:type="string" ovf:value="${appserver-vm}">
  <Description>IP address of the application server VM</Description>
</Property>
</ProductSection>
```

需要注意的一点是，在 ovf:key 属性的取值中不得带有句号（“.”）或者冒号（“:”）。在 ovf:class 和 ovf:instance 属性的取值中不得带有冒号（“:”）。

如果只安装了一个产品实例，则不需要使用 ovf:instance 属性。

下面这个示例中描述了 ProductSection（产品字段）元素中 OVF 属性的使用情况：

```
<ProductSection>
  <Property ovf:key="adminEmail" ovf:type="string" ovf:userConfigurable="true"
    ovf:configuration="standard">
    <Label>Admin email</Label>
    <Description>Email address of service administrator</Description>
  </Property>
  <Property ovf:key="appLog" 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>
```

在上面的示例中，在标准配置模式下，adminEmail（管理员电子邮件）属性是唯一可以由用户配置的属性。在最低配置模式下，appLog 属性的默认值从“low”变为了“none”。

### 4.5.3 EulaSection 元素

EulaSection（最终用户许可协议字段）元素中含有人类可读的授权协议，通常是 VirtualSystem（虚拟系统）或者 VirtualSystemCollection（虚拟系统集合）元素。每个母元素中都包括有多个 EulaSection（最终用户许可协议字段）元素。在 OVF 包部署之后，每个 VirtualSystemCollection（虚拟系统集合）元素中的许可（License）元素都将展示给客户，供客户进行阅读和确认。如果支持无人部署功能，则必须规定隐性接受



Error! Unknown document property name.

## 开放虚拟化格式白皮书

最终用户许可协议的功能。

最终用户许可协议可以外部化、本地化、或者指向外部的许可文件。更多关于国际化方面的信息，参阅本文件第 5.2 节相关内容。

下面是一个 EulaSection（最终用户许可协议字段）元素示例：

```
<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 dui. Enim
    eros in vel, volutpat nec pellentesque leo, scelerisque.
  </License>
</EulaSection>
```

### 4.5.4 VirtualHardwareSection 元素

VirtualHardwareSection（虚拟硬件字段）元素中描述了使用 CIM 资源配置设置数据模型的虚拟硬件。此模型基于 CIM\_ResourceAllocationSettingData（CIM\_资源配置设置数据）分级，规定了用于描述所请求资源的类型和数量的 CIM 属性。CIM 架构可以通过下列网址获得：<http://www.dmtf.org/standards/cim>。

在下面这个示例中，描述了虚拟处理器设备所需要的最低要求资源配置设置数据。这些数据不具有用户友好性，因此加入了 rasd:Description（描述）和 rasd:ElementName（元素名称）属性。

```
<Item>
  <rasd:InstanceID>0</epasd:InstanceID>
  <rasd:ResourceType>3</epasd:ResourceType>
</Item>
```

此 VirtualHardwareSection（虚拟硬件字段）元素描述了虚拟系统使用的虚拟装置的硬件抽象层。CIM\_ResourceAllocationSettingData（CIM\_资源配置设置数据）分级中带有装置列表。部分装置，例如以太网端口和存储器可以进一步使用扩展属性组进行分级。

在此具体示例中，规定了一套典型的硬件组（500MB 的内存、一台 CPU、一台 NIC、以及一个虚拟磁盘）。在这里引用了外部字段中规定的网络和磁盘识别符。硬件字段不完整或者缺失会导致部署失败。

下面这个示例中描述了 VirtualHardwareSection（虚拟硬件字段）元素。

```
<VirtualHardwareSection>
  <Info>Memory = 4 GB, CPU = 1 GHz, Disk = 100 GB, 1 Ethernet nic</Info>
```

```

<Item>
  <rasd:AllocationUnits>Hertz*10^9</rasd:AllocationUnits>
  <rasd:Description>Virtual CPU</rasd:Description>
  <rasd:ElementName>1 GHz virtual CPU</rasd:ElementName>
  <rasd:InstanceID>1</rasd:InstanceID>
  <rasd:Reservation>1</rasd:Reservation>
  <rasd:ResourceType>3</rasd:ResourceType>
  <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
  <rasd:VirtualQuantityUnit>Count</ rasd:VirtualQuantityUnit>
</Item>
<Item>
  <rasd:AllocationUnits>byte*2^30</rasd:AllocationUnits>
  <rasd:Description>Memory</rasd:Description>
  <rasd:ElementName>1 GByte of memory</rasd:ElementName>
  <rasd:InstanceID>2</rasd:InstanceID>
  <rasd:Limit>4</rasd:Limit>
  <rasd:Reservation>41</rasd:Reservation>
  <rasd:ResourceType>4</rasd:ResourceType>
</Item>
<EthernetPortItem>
  <epasd:AllocationUnits>bit / second *2^30 </rasd:AllocationUnits>
  <epasd:Connection>VM Network</epasd:Connection>
  <epasd:Description>Virtual NIC</epasd:Description>
  <epasd:ElementName>Ethernet Port</epasd:ElementName>
  <epasd:NetworkPortProfileID>1</epasd:NetworkPortProfileID>
  <epasd:NetworkPortProfileIDType>4</epasd:NetworkPortProfileIDType>
  <epasd:ResourceType>10</epasd:ResourceType>
  <epasd:VirtualQuantity>1</epasd:VirtualQuantity>
  <epasd:VirtualQuantityUnits>Count</epasd:VirtualQuantityUnits>
</EthernetPortItem>
<StorageItem>
  <sasd:AllocationUnits>byte*2^30</sasd:AllocationUnits>
  <sasd:Description>Virtual Disk</sasd:Description>
  <sasd:ElementName>100 GByte Virtual Disk</sasd:ElementName>

  <sasd:Reservation>100</sasd:Reservation>
  <sasd:ResourceType>31</sasd:ResourceType>
  <sasd:VirtualQuantity>1</sasd:VirtualQuantity>

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<rasd:VirtualQuantityUnit>Count</rasd:VirtualQuantityUnit>
</StorageItem>
</VirtualHardwareSection>
```

ResourceSubType（资源子类型）CIM 属性的示例如下所述：

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

下面这个示例里描述了带有默认值和“big”取值的 VirtualHardwareSection（虚拟硬件字段）属性。更多关于如何使用配置选项的信息，参阅第 4.4.6 节相关内容。

```
<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>
```

### 4.5.5 PlacementSection 元素

PlacementSection（布置字段）元素中规定了虚拟系统或虚拟系统集合是其中一个成员的布置组。部署功能可以调用布置组中规定的布置政策（参阅第 4.4.8 节相关内容）。下面的 OVF 描述文件片段中描述了两套虚拟系统中的布置字段。

```
<VirtualSystemCollection ovf:id="VSC10">
  <VirtualSystem ovf:id="VS11">
    <Info>Web server</Info>
    ...
```

Error! Unknown document property name.

```
<ovf:PlacementSection ovf:group="PG1">
  <Info>Placement policy group reference</Info>
</ovf:PlacementSection>
...
</VirtualSystem>
<VirtualSystem ovf:id="VS21">
  <Info>Web server</Info>
  ...
  <ovf:PlacementSection ovf:group="PG1">
    <Info>Placement policy group reference</Info>
  </ovf:PlacementSection>
  ...
</VirtualSystem>
</VirtualSystemCollection>
```

在此示例中，在虚拟系统实例化之后，将会根据“PG1”布置组中规定的布置政策进行布置。从本文件第 4.4.8 节中可以看出，“PG1”布置组具有亲和性布置政策。图 7 中列出了亲和性布置的基本情况。

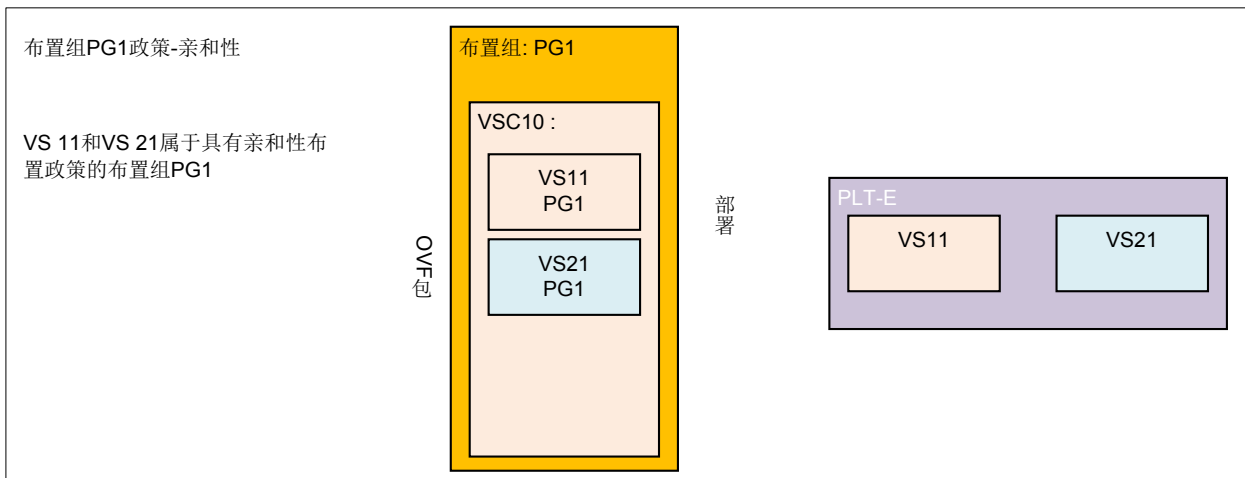


图 7: 亲和性布置

下面这个 OVF 描述文件片段中描述了两套虚拟系统的布置字段:

```
<VirtualSystemCollection ovf:id="VSC10">
  <VirtualSystem ovf:id="VS11">
    <Info>Web server</Info>
    ...
    <ovf:PlacementSection ovf:group="PG2">
      <Info>Placement policy group reference</Info>
    </ovf:PlacementSection>
    ...
  </VirtualSystem>
  ...
</VirtualSystemCollection>
```

```
</VirtualSystem>

<VirtualSystem ovf:id="VS21">
  <Info>Web server</Info>
  ...
  <ovf:PlacementSection ovf:group="PG2">
    <Info>Placement policy group reference</Info>
  </ovf:PlacementSection>
  ...
</VirtualSystem>
</VirtualSystemCollection>
```

在此示例中，在虚拟系统实例化之后，将会根据“PG2”布置组中规定的布置政策进行布置。从本文件第 4.4.8 节中可以看出，“PG2”布置组具有可用性布置政策。图 8 中列出了可用性布置的基本情况。

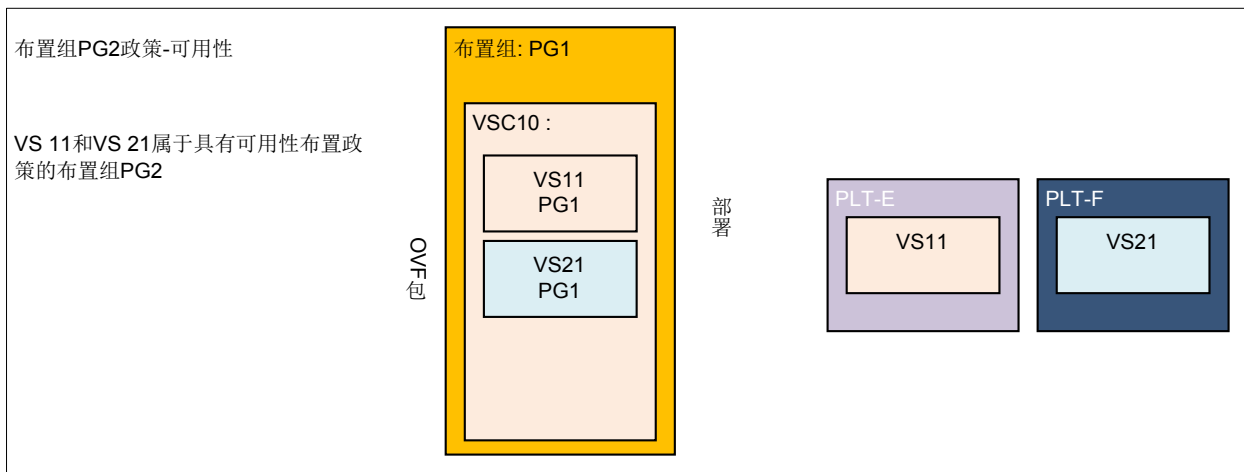


图 8: 可用性布置

下面这个 OVF 描述文件片段中描述了两套虚拟系统的布置字段:

```
<VirtualSystemCollection ovf:id="VSC10">
  <ovf:PlacementSection ovf:group="PG1">
    <Info>Placement policy group reference</Info>
  </ovf:PlacementSection>

  <VirtualSystem ovf:id="VS11">
    <Info>Web server</Info>
    ...
  </VirtualSystem>

  <VirtualSystem ovf:id="VS21">
    <Info>Web server</Info>
    ...
  </VirtualSystem>
</VirtualSystemCollection>
```

Error! Unknown document property name.

```
<ovf:PlacementSection ovf:group="PG2">
  <Info>Placement policy group reference</Info>
</ovf:PlacementSection>
...
</VirtualSystem>
<VirtualSystem ovf:id="VS31">
  <Info>Web server</Info>
  ...
</VirtualSystem>
<VirtualSystem ovf:id="VS41">
  <Info>Web server</Info>
  ...
  <ovf:PlacementSection ovf:group="PG2">
    <Info>Placement policy group reference</Info>
  </ovf:PlacementSection>
  ...
</VirtualSystem>
</VirtualSystemCollection>
```

在此示例中，在虚拟系统实例化之后，将会根据“PG1”和“PG2”布置组中规定的布置政策进行布置。从本文件第 4.4.8 节中可以看出，“PG1”布置组具有亲和性布置政策，“PG2”布置组具有可用性布置政策。图 9 中描述了这个更加复杂的示例的虚拟系统的布置情况。该图中还描述了传递规则的使用情况。

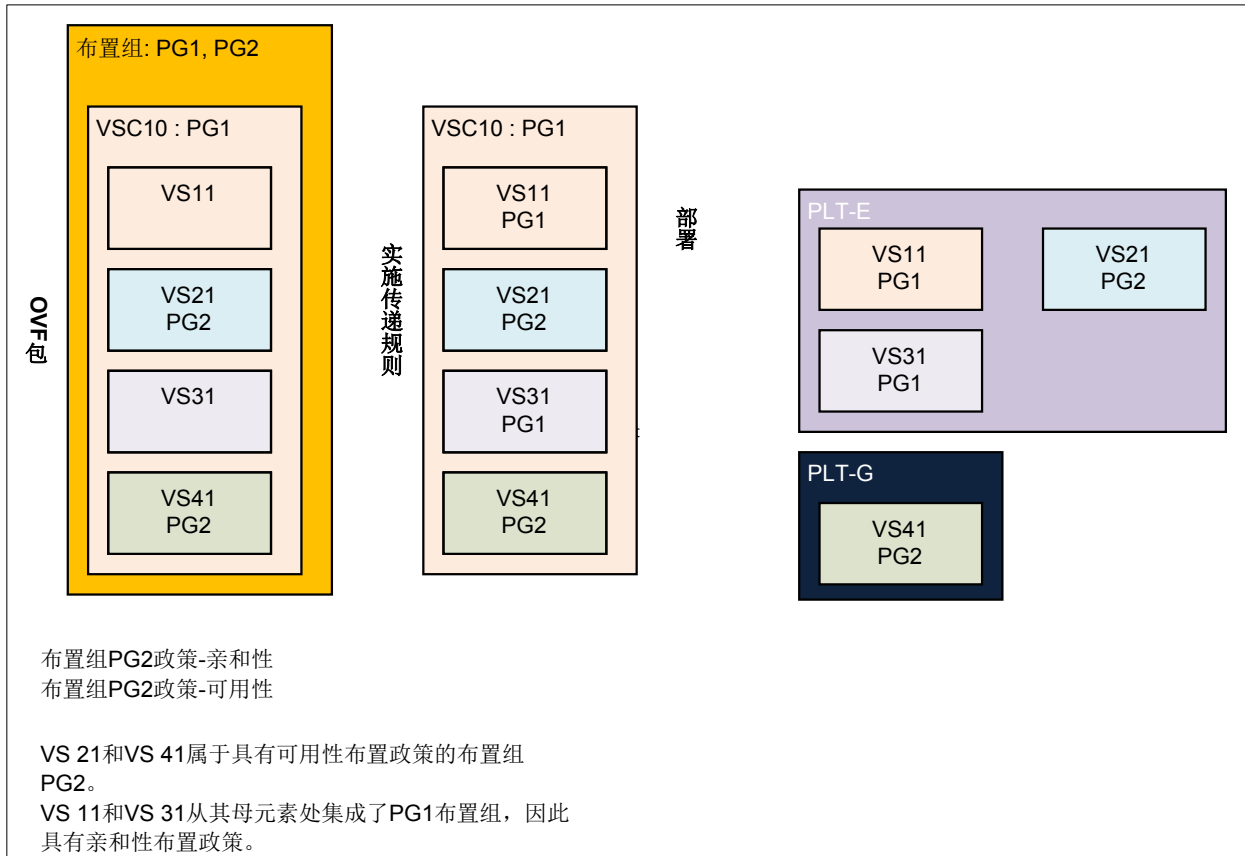


图 9: 亲和性和可用性布置

#### 4.5.6 EncryptionSection 元素

在启用 OVF 设备互换的过程中需要使用加密机制有着多个原因, 但主要目的都是确保只有预计用户能够使用相关虚拟应用。本规范中建议采用的加密机制利用了现有的加密标准, 这些加密标准将被纳入到本规范规定功能中。

EncryptionSection (加密字段) 元素中提供了一个位置, 用于放置加密算法相关的标记, 以及指向已经加密的 OVF 内容的相关引用列表。

一个文档通常只会使用一种加密方法, 具有单个密钥。尽管如此, 本规范在对 OVF 描述文件的不同部分进行加密处理时具有一定的灵活性, 可以使用不同的方法提供不同的密钥, 并以不同的方式与最终用户进行通信。

非常重要的一点是, 应该随时注意到取决于对 OVF 描述文件中哪些部分进行了加密处理, 在解密之前, 部分 OVF 描述文件可能无法使用 OVF 根据进行验证。

使用 XML 加密标准 1.1 进行加密处理, 可以对引用字段中的文件进行加密, 也可以对 OVF 文档中的 XML 标记的任何部分进行加密。

从加密的角度看, 本标准中规定的重要内容包括:

- a) 用于衍生在加密过程中使用的任何密钥的算法;

Error! Unknown document property name.

- b) 用于对使用密钥的内容进行加密处理的模块加密算法；
- c) 用于传输嵌入到 OVF XML 文档中的密钥的方法。

对于文档中使用的每种加密方法而言，所有这些内容都需要由 OVF 包作者进行定义。例如，作者可以嵌入文档使用的密钥，或者作者也可以选择通过其他途径将密钥告知最终用户。

其他方面的问题包括在 OVF 包络中的引用字段中加入引用列表，或者使用具体的方法对引用字段中的文件进行加密处理。为了确保能够对 OVF 描述文件中的任意字段进行加密，应该使用 ReferenceList（引用列表）元素中的 XML ID 属性。

下面的示例中解释了 Encryption（加密）字段的概念结构：

```
<!-- Start of encryption section --!>
  <!-- Start of Markup for encryption method 1 --!>
    <!-- Markup defining key derivation aspects per XML encryption 1.1 --!>
    <!-- Markup defining the usage of the key for encryption per XML encryption 1.1 --!>
    <!-- Optionally, the markup for key transportation per XML encryption 1.1 --!>
    <!-- Start of markup for pointers to the list of XML fragments encrypted using method 1 --!>
      <!-- Pointer 1 --!>
        .
        .
      <!-- Pointer N --!>
    <!-- End of markup for pointers to the list of XML fragments encrypted using method 1 --!>
  <!-- End of Markup for method 1 of encryption --!>

  <!-- Start of the markup for encryption method N --!>
    <!-- Markup defining key derivation aspects per XML encryption 1.1 --!>
    <!-- Markup defining the usage of the key for encryption per XML encryption 1.1 --!>
    <!-- Optionally, the markup for key transportation per XML encryption 1.1 --!>
    <!-- Start of markup for pointers to the list of XML fragments encrypted using method 1 --!>
      <!-- Pointer 1 --!>
        .
        .
      <!-- Pointer N --!>
    <!-- End of Markup for encryption method N --!>
<!-- End of encryption section --!>
```



Error! Unknown document property name.

## 开放虚拟化格式白皮书

在下面这个示例中，提供了在 OVF 文档中使用的加密方法和 OVF 加密字段。另外，指向本项目的相关的引用列表也进行了加密处理。

```
<ovf:EncryptionSection>
<!-- This section contains two different methods of encryption and the corresponding
backpointers to the data that is encrypted. -->
  <!-- Method#1: Pass phrase based key derivation -->
<!-- The following derived key block defines PBKDF2 and the corresponding backpointers to
the encrypted data elements. -->
  <!-- Use a salt value "ovfpassword" and iteration count of 4096. --->
<xenc11:DerivedKey>
  <xenc11:KeyDerivationMethod
Algorithm="http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-5#pbkdf2"/>
<pkcs-5:PBKDF2-params>
  <Salt>
    <Specified>ovfpassword</Specified>
  </Salt>
  <IterationCount>4096</IterationCount>
  <KeyLength>16</KeyLength>
  <PRF Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256"/>
  </pkcs-5:PBKDF2-params>
...
<!-- The ReferenceList element below contains references to the file Ref-109.vhd via the URI
syntax that is specified by XML Encryption.
--->
<xenc:ReferenceList>
  <xenc:DataReference URI="#first.vhd" />
<xenc:DataReference URI=... />
<xenc:DataReference URI=... />
</xenc:ReferenceList>
  </xenc11:DerivedKey>
  <!-- Method#2: The following example illustrates use of a symmetric key transported by
using the public key within a certificate. -->
<xenc:EncryptedKey>
  <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'
  <ds:X509Data>
    <ds:X509Certificate> ... </ds:X509Certificate>
  </ds:X509Data>
```

Error! Unknown document property name.

```

    </ds:KeyInfo>
    <xenc:CipherData>
    <xenc:CipherValue> ... </xenc:CipherValue>
    </xenc:CipherData>
<!-- The ReferenceList element below contains reference #second-xml-fragment" to the XML
fragment that has been encrypted by using the above method. --->
    <xenc:ReferenceList>
        <xenc:DataReference URI='#second-xml-fragment' />
        <xenc:DataReference URI='...' />
        <xenc:DataReference URI='...' />
    </xenc:ReferenceList>
</xenc:EncryptedKey>
</ovf:EncryptionSection>

```

Below is an example of the encrypted file that is referenced in the EncryptionSection above by using URI='Ref-109.vhd' syntax.

EXAMPLE:

```

<ovf:References>
<ovf:File
ovf:id="Xen:9cb10691-4012-4aeb-970c-3d47a906bfff/0b13bdba-3761-8622-22fc-2e252ed9ce14"
ovf:href="Ref-109.vhd">
<!-- The encrypted file referenced by the package is enclosed by an EncryptedData with a
CipherReference to the actual encrypted file. The EncryptionSection in this example has a
backpointer to it under the PBKDF2 algorithm via Id="first.vhd". This tells the decrypter how
to decrypt the file. -->
<xenc:EncryptedData Id="first.vhd" Type='http://www.w3.org/2001/04/xmlenc#Element' >
    <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
    <xenc:CipherData>
        <xenc:CipherReference URI='Ref-109.vhd' />
    </xenc:CipherData>
</xenc:EncryptedData>
</ovf:File>
</ovf:References>

```

Below is an example of the encrypted OVF markup that is referenced in the EncryptionSection above by using URI='#second-xml-fragment' syntax.

EXAMPLE:

```

<!-- The EncryptedData element below encompasses encrypted xml from the original document.
It is provided with the Id "second-xml-fragment", which allows it to be referenced from the
EncryptionSection. -->
<xenc:EncryptedData Type=http://www.w3.org/2001/04/xmlenc#Element

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
Id="second-xml-fragment">
<!-- Each EncryptedData specifies its own encryption method. -->
  <xenc:EncryptionMethod Algorithm=http://www.w3.org/2001/04-xmenc#aes128-cbc/>
  <xenc:CipherData>
    <!-- Encrypted content -->
    <xenc:CipherValue>DEADBEEF</xenc:CipherValue>
  </xenc:CipherData>
</xenc:EncryptedData>
```

## 4.6 虚拟系统集合中使用的 OVF 字段元素

### 4.6.1 ResourceAllocationSection 元素

ResourceAllocationSection（资源配置字段）元素中规定了对虚拟系统集合的资源限制。相反，VirtualHardwareSection（虚拟硬件字段）元素适用于具体的虚拟系统。

ResourceAllocationSection（资源配置字段）元素可以使用 bound（范围）属性来设置向虚拟系统集合中的所有虚拟系统配置的资源的最小值、最大值，或者同时对两者做出规定。

下面的示例中对 ResourceAllocationSection（资源配置字段）元素进行了描述，其中处理器配置中使用了 bound（范围）属性：

```
<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: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>
  <Item ovf:bound="max">
    <rasd:AllocationUnits>MHz</rasd:AllocationUnits>
    <rasd:ElementName>1500 MHz reservation</rasd:ElementName>
```

Error! Unknown document property name.

```
<rasd:InstanceID>1</rasd:InstanceID>
<rasd:Reservation>1500</rasd:Reservation>
<rasd:ResourceType>3</rasd:ResourceType>
</Item>
</ResourceAllocationSection>
```

### 4.6.2 StartupSection 元素

StartupSection（启动字段）元素中含有控制虚拟系统集合启动和关闭的措施，在 InstallSection（安装字段）元素之后执行。StartupSection（启动字段）元素是一个 Item（项目）元素的列表。Item（项目）元素中带有能控制开机关机顺序和时机的属性。StartupSection（启动字段）元素中的 Item（项目）元素包括在该元素范围内。不得将该元素与 VirtualHardwareSection（虚拟硬件字段）元素或者 ResourceAllocationSection（资源配置字段）元素中的 Item（项目）元素混淆起来。

StartupSection（启动字段）元素中的 Item（项目）元素可以引用 VirtualSystem（虚拟系统）元素或者 VirtualSystemCollection（虚拟系统集合）元素。StartupSection（启动字段）元素可以控制包括在虚拟系统母集中的虚拟系统和虚拟系统集合的启动和关闭。相关格式允许采用递归式启动结构。参阅图 10 相关内容。

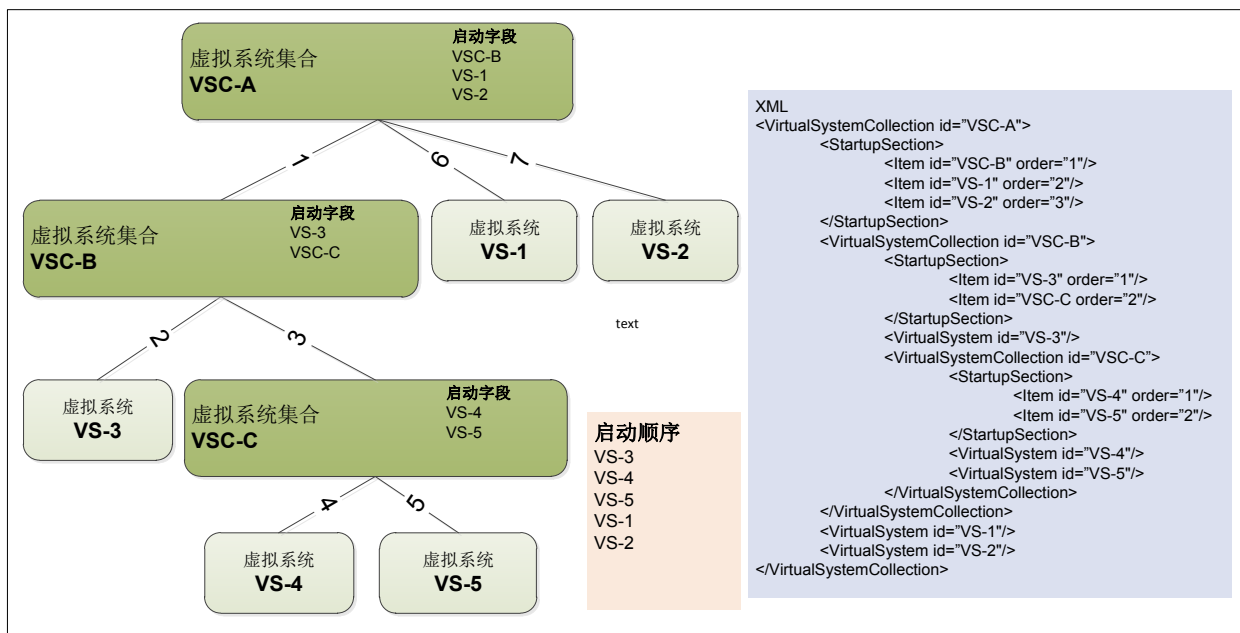


图 10: StartupSection（启动字段）遍历

启动顺序通过 VirtualSystemCollection（虚拟系统集合）元素中的 StartupSection（启动字段）元素中的 Item（项目）元素的 order（顺序）属性做出规定。order（顺序）属性是一个不可以取负值的整数。

如果 Item（项目）元素的 order（顺序）属性的取值为“0”（零），则说明虚拟系统可以随时启动。虚拟平台不需要等待具有更高 order（顺序）属性取值的项目先启动。

如果 Item（项目）元素的 order（顺序）属性的取值为非零数值，则虚拟系统应该按照数值的升序进行启动。具有相同的 Item（项目）元素的 order（顺序）属性取值的虚拟系统将会同时启动。

Error! Unknown document property name.

## 开放虚拟化格式白皮书

建议按照数字的降序顺序对虚拟系统进行停机处理，如果 Item（项目）元素的 order（顺序）属性的取值为“0”（零），则说明虚拟系统可以随时停机。

尽管如此，在具体实施过程中，除非规定了 shutdownorder（关机顺序）属性，否则也允许不按照降序顺序对虚拟系统进行停机。通过 shutdownorder（关机顺序）属性可以规定系统的具体停机顺序。

在 Item（项目）元素中还包括有多种可选的属性，可以用于更加具体地控制启动和停机过程。其中 startDelay（启动延迟）和 stopDelay（停机延迟）属性规定了在执行流程中的下一个步骤时需要等待的时间。startDelay（启动延迟）和 stopDelay（停机延迟）属性的默认取值都为零。

startAction（启动动作）和 stopAction（停机动作）属性中规定了在启动和停机过程中需要使用的动作类型。startAction（启动动作）属性的有效取值为“powerOn（启动）”和“none（无）”，默认取值为“powerOn（启动）”。stopAction（停机动作）属性的有效取值为“powerOn（启动）”、“guestShutdown（客户端关机）”和“none（无）”，默认取值为“guestShutdown（客户端关机）”，具体采取的动作随部署平台的不同而不同。

waitingForGuest（等待客户端）属性是一个布尔值属性，允许部署平台在客户端软件报告准备就绪之前等待一段时间。其默认值为“FALSE”。所采用的通信机制随部署平台的不同而不同。

下面这个示例中描述了 StartupSection（启动字段）元素：

```
<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>
```

### 4.6.3 ScaleOutSection 元素

ScaleOutSection（扩展字段）元素允许对 VirtualSystemCollection（虚拟系统集合）元素中实例化的虚拟系统的数量进行动态配置。如果 VirtualSystemCollection（虚拟系统集合）元素中没有 ScaleOutSection（扩展字段）元素，则虚拟系统和虚拟系统集合的数量将被固定。ScaleOutSection（扩展字段）元素将规定可以创建的复型的最大和最小数量。在部署之时，部署平台可以选择一个位于 InstanceCount（实例计数）最大值和最小值之间的数值。用户可以查询该数值，或者部署平台也可以根据其他元数据确定该数值。ScaleOutSection（扩展字段）元素只会出现在 VirtualSystemCollection（虚拟系统集合）元素中，尽管虚拟系统和虚拟系统集合都可以被复制。

下面这个示例描述了 ScaleOutSection（扩展字段）元素：

```
<VirtualSystemCollection ovf:id="web-tier">
  ...
  <ovf:ScaleOutSection ovf:id="web-server">
```

Error! Unknown document property name.

```

<Info>Web tier</Info>

<ovf:Description>Number of web server instances in web tier</ovf:Description>

<ovf:InstanceCount ovf:default="4" ovf:minimum="2" ovf:maximum="8"/>

</ovf:ScaleOutSection>

...

<VirtualSystem ovf:id="web-server">

  <Info>Prototype web server</Info>

  ...

</VirtualSystem>

</VirtualSystemCollection>

```

在上面这个示例中，部署平台创建了一个含有两到八个网页服务器虚拟机实例的网页层，其默认数值为四。这样，部署平台就可以做出适宜的选择（例如：通过提示用户进行选择）。假设创建了三个复型，则第一个复型的客户端软件可以使用的 OVF 环境应该具有如下所述的内容：

```

<Environment ... ovfenv:id="web-server-1">

...

<Entity ovfenv:id="web-server-2">

...

</Entity>

<Entity ovfenv:id="web-server-3">

...

</Entity>

</Environment>

```

需要注意的一点是，复型的 OVF ID 来自于虚拟机原型机的 ID，再加上一个序列号。在部署之后，所有复型虚拟系统都具有一个序列号后缀，没有任何虚拟系统使用原型机的基础 ID。如果存在 StartupSection（启动字段）元素，则每个复型都应该具有相同的启动顺序值。不可以为复型规定不同的启动顺序。

示例：

```

<VirtualSystemCollection ovf:id="web-tier">

...

<DeploymentOptionSection>

  <Info>Deployment size options</Info>

  <Configuration ovf:id="minimal">

    <Label>Minimal</Label>

    <Description>Minimal deployment scenario</Description>

  </Configuration>

  <Configuration ovf:id="common" ovf:default="true">

    <Label>Typical</Label>

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<Description>Common deployment scenario</Description>
</Configuration>
...
</DeploymentOptionSection>
...
<ovf:ScaleOutSection ovf:id="web-server">
  <Info>Web tier</Info>
  <ovf:Description>Number of web server instances in web tier</ovf:Description>
  <ovf:InstanceCount ovf:default="4"/>
  <ovf:InstanceCount ovf:default="1" ovf:configuration="minimal"/>
</ovf:ScaleOutSection>
...
</VirtualSystemCollection>
```

在上面这个示例中，使用 `DeploymentOptionSection`（部署选项字段）元素来控制 `ScaleOutSection`（扩展字段）元素中的 `InstanceCount`（实例计数）元素的取值。`ScaleOutSection`（扩展字段）元素的取值还可以通过 `property`（属性）元素进行控制。对于每个复型而言，都将提示一次 OVF 属性。如果作者希望 OVF 属性在不同复型之间共享，则可以将其放入到 `VirtualSystemCollection`（虚拟系统集合）元素中。

## 4.7 虚拟系统中使用的 OVF 字段元素

### 4.7.1 OperatingSystemSection 元素

`OperatingSystemSection`（操作系统字段）元素中规定了在虚拟系统中，客户端使用的操作系统。可以通过 `CIM_OperatingSystem.OSType` 属性来选择操作系统。OVF Version（版本）和 OVF ID 将与该属性的 Value（取值）和 ValueMap（取值映射）相对应。

ID 属性是强制要求属性，其取值应该为 ValueMap（取值映射）中的整数。Version（版本）属性是可选属性，与 ValueMap（取值映射）中的 Value（取值）相对应。Version（版本）属性是一个符号字符串，不能进行国际化。

Info（信息）（来自于字段）和 Description（描述）元素都可以外部化和本地化，具体参阅第 5.2 节相关内容。

下面这个示例是规定采用 Microsoft Windows Server 2008 操作系统的字段：

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

### 4.7.2 InstallSection 元素

InstallSection（安装字段）元素是一个可选元素，只能用在 VirtualSystem（虚拟系统）元素中。在出现此元素时，其将在 StartupSection（启动字段）元素之前被处理。

InstallSection（安装字段）元素可以让 OVF 包作者规定在停机之前需要重启的虚拟系统，以便完成 OVF 包的安装。在典型情况下，在引导过程中，客户端软件需要执行脚本、或者 OVF 环境中的其他软件以完成安装。如果没有 InstallSection（安装字段）元素，则说明无需引导过程就可以完成安装。例如，如果虚拟系统中没有客户端软件、或者客户端软件安装在系统镜像中，则不需要 InstallSection（安装字段）元素。

虚拟系统集中的每套虚拟系统都可以规定一个 InstallSection（安装字段）元素，这些虚拟系统可以同时重启。

InitialBootStopDelay（初始引导停机延迟）属性的取值是在虚拟系统关机之前，虚拟平台需要等待的时间，单位为秒。如果在超出上述属性取值的时间后，虚拟系统依然没有关机，则应该视为安装已经失败。

InitialBootStopDelay（初始引导停机延迟）属性的默认值为零，这就意味着对于延迟时间没有限制，虚拟平台将一直等待，直到虚拟系统自行停机为止。在停机之前，虚拟系统中的客户端软件不能多次引导。

在下面的示例中，在客户端软件关闭虚拟系统之前，虚拟平台需要等待五分钟（300 秒时间）。如果虚拟机没有在五分钟时间内关机，则安装视为失败。在五分钟的等待时间内，虚拟系统可以多次重启。

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

### 4.7.3 EnvironmentFilesSection 元素

EnvironmentFilesSection（环境文件字段）元素能够向客户端软件传输额外的环境文件，以便进行自定义操作。这些文件使用预 OVF 环境文件相同的传输媒介进行传输。

OVF 环境文件通过部署功能生成。尽管如此，其他环境文件将不能通过部署功能生产，必须由 OVF 包作者提供。其他环境文件将在 EnvironmentFilesSection（环境文件字段）元素中做出规定，该元素中包括一个 File（文件）元素，每个文件都带有 ovf:fileRef 属性和 ovf:path 属性。

ovf:fileRef 属性指向 References（引用）元素中的一个文件元素。通过比较其 ovf:id 属性取值与 ovf:fileRef 属性取值，可以确定 File（文件）元素的身份并进行识别。

ovf:path 属性描述了文件在所属的传输媒介中的相对位置。

```
<Envelope>
  <References>
    ...
  <File ovf:id="config" ovf:href="config.xml" ovf:size="4332"/>
```



Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<File ovf:id="resources" ovf:href="http://mywebsite/resources/resources.zip"/>
</References>
...
<VirtualSystem ovf:id="...">
  ...
  <ovf:EnvironmentFilesSection ovf:required="false" ovf:transport="iso">
    <Info>Config files to be included in OVF environment</Info>
    <ovf:File ovf:fileRef="config" ovf:path="setup/cfg.xml"/>
    <ovf:File ovf:fileRef="resources" ovf:path="setup/resources.zip"/>
  </ovf:EnvironmentFilesSection>
  ...
</VirtualSystem>
...
</Envelope>
```

在上面这个示例中，OVF 包中的 `config.xml` 被复制到了 OVF 环境 ISO 镜像中，可以通过客户端软件在 `/ovffiles/setup/cfg.xml` 位置处访问，而文件 `resources.zip` 可以在部署之时通过 `/ovffiles/setup/resources.zip` 路径访问。

### 4.7.4 BootDeviceSection 元素

早期版本的 OVF 允许虚拟系统只通过默认的引导设备进行引导。尽管如此，后来发现这对于在 OVF 部署过程中可能遇到的很多情况而言是一种限制因素。

- a) 没有任何方式可以规定虚拟系统是否需要设置为通过 NIC 进行 PXE 引导。同样的，也没有任何方式可以规定虚拟系统是否需要设置为从辅助磁盘或者 USB 装置进行引导。因此，存在对规定替代引导来源及其相关设置的能力的需求。
- b) 通过具体实施经验还发现了其他需求，需要能够规定多个替代引导配置项。例如，在 OVF 的“准备”阶段，可能需要使用修复磁盘对虚拟系统进行更新。

通用信息模型（CIM）中规定了行业中普遍采用的，针对台式机和服务器的 BIOS 的处理引导顺序用途实例的属性。其中 `CIM_BootSourceSetting`（CIM\_引导来源设置）中规定了可以使用具体的引导来源设备，例如 NIC 或者磁盘来作为引导源。每个设备都将通过独特的 ID 进行识别，这些 ID 将在 `CIM_BootSourceSetting`（CIM\_引导来源设置）中进行规定。

在 `CIM_BootConfigSetting`（CIM\_引导配置设置）分级中，可以通过规定引导设备的顺序，对引导过程进行配置。因此，可以将含有一个或多个 `CIM_BootSourceSetting`（CIM\_引导来源设置）属性的引导顺序整合到 `CIM_BootConfigSetting`（CIM\_引导配置设置）分级中。

OVF envelope（包络）元素也允许将多个上述引导配置项整合到 `BootDeviceSection`（引导设备字段）元素中。每个 `BootDeviceSection`（引导设备字段）元素都是 `VirtualHardwareSection`（虚拟硬件字段）元素的一

部分。

部署功能可以根据其选择的引导配置的规定，设置虚拟系统的引导来源顺序。如果存在多种引导配置，则选择正确的引导配置是非常重要的。部署功能可以根据部署的状态、以及引导配置结构中的重要元素来进行上述选择。

在下面的示例中，安装前配置项中将引导来源规定为具体的设备（网络），而安装后配置项将其规定为设备类型（硬盘）：

```
EXAMPLE:
<Envelope>
...
<VirtualSystem ovf:id="...">
...
<ovf:BootDeviceSection>
  <Info>Boot device order specification</Info>
  <bootc:CIM_BootConfigSetting>
    <bootc:Caption>Pre-Install</bootc:Caption>
    <bootc:Description>Boot Sequence for fixup of disk</bootc:Description>
    <boots:CIM_BootSourceSetting>
      <boots:Caption>Fix-up DVD on the network</boots:Caption>
      <boots:InstanceID>3</boots:InstanceID>          <!-- Network device-->
    </boots:CIM_BootSourceSetting>
    <boots:CIM_BootSourceSetting>
      <boots:Caption>Boot virtual disk</boots:Caption>
      <boots:StructuredBootString>CIM:Hard-Disk</boots:StructuredBootString>
    </boots:CIM_BootSourceSetting>
  </bootc:CIM_BootConfigSetting>
</ovf:BootDeviceSection>
...
</VirtualSystem>
</Envelope>
```

## 5. OVF 包授权

### 5.1 创建

OVF 创建包括下列步骤：

Error! Unknown document property name.

## 开放虚拟化格式白皮书

- i) 将一套 VM 打包到一组虚拟磁盘中；
- ii) 对这些虚拟磁盘进行适宜的编码；
- iii) 在虚拟硬件的规范、许可、以及其他自定义元数据中，加入一个 OVF 描述文件。
- iv) 可以在附件中包括一个 OVF 包的数字签名。

部署 OVF 包的流程是指由虚拟平台使用 OVF 包，并使用其内容创建一组虚拟机。

创建 OVF 的流程可以非常简单，可以直接将现有的虚拟机从虚拟平台导出到 OVF 包中，并在其中加入必要的元数据，以确保其可以正确安装和执行即可。通过此流程可以将特定管理程序上运行的虚拟机的运行时状态转化为 OVF 包。在此流程中，可以将虚拟机的磁盘进行压缩处理，确保其能够更便捷地进行发布。

对于商业级别的虚拟机而言，可以使用标准构建环境来生成 OVF 包。例如，可以使用源控制系统来管理 OVF 描述文件，同时可以使用可复制的脚本文件（例如 make 文件）、或者通过使用由不同供货商的设备构建工具包来构建 OVF 包。

在 OVF 包创建完毕之后，可以在其中加入虚拟应用具体的、安装后的配置元数据。其中包括可选的虚拟应用界面语言本地化元数据，最终用户许可协议的查看/签署以及/或者执行，以及资源配置方面的元数据。还可以在客户端中加入特殊的驱动器、代理、以及其他工具，以便增强（例如）输入/输出、计时、内存管理、监控、以及按顺序停机等功能。

授权 OVF 描述文件的流程从本质上看是将构建虚拟应用的模块放置在一起的过程。正如本文前文中相关内容所述，虚拟应用可以通过构成设备的虚拟系统的描述、关于虚拟应用和客户端软件的元数据、以及一组引用文件进行定义。OVF 描述文件是整合和引用所有必要信息的核心元素。OVF 描述文件的主要构件模块是字段。在本文件第 4 章中，介绍了可以用于描述各种虚拟应用的字段。

## 5.2 国际化

OVF 规范支持使用可选的 `ovf:msgid` 属性来进行消息本地化处理。在部署过程中，经过本地化处理的消息可以使用本地语言向用户提供必要信息。

```
<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">
```

Error! Unknown document property name.

```
<Msg ovf:msgid="info.os">Betriebssystem</Msg>
...
</Strings>
</Envelope>
```

上面这个示例中定义了字段内部的 Info(信息)元素。此字段中的信息与虚拟系统的操作系统相关。ovf:msgid 属性的取值为“info.os”，这就意味着 Info（信息）元素中，起始标签和结束标签之间的字符串可以使用本地化语言进行替换。本地化的消息将通过其在 info.os 中规定的消息 ID 进行引用。如果 Strings（字符串）字段中有合适的本地化消息，则默认消息“Operating System（操作系统）”将会被对当前地区对应的 Strings（字符串）字段中包含的本地化消息替代。

在上面那个示例中，本地化字符串保存在 OVF 描述文件内。本地化字符串还可以通过使用外部的字符串包，保存在 OVF 描述文件的外部。例如：

```
<Envelope ...>
  <References>
    ...
    <File ovf:id="da-DA-resources" ovf:href="danish.msg"/>
    <File ovf:id="de-DE-resources" ovf:href="german.msg"/>
    ...
  </References>
  ...
  <Info ovf:msgid="info.os">Operating System</Info>
  ...
  <Strings xml:lang="da-DA" ovf:fileRef="da-da-resources"/>
  <Strings xml:lang="de-DE" ovf:fileRef="de-de-resources"/>
</Envelope>
```

在 danish.msg 和 german.msg 文件中规定了“Operating System（操作系统）”的本地化消息。外部消息文件 german.msg 将在下面这个示例中进行描述：

```
<Strings
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns="http://schemas.dmtf.org/ovf/envelope/1"
  xml:lang="de-DE">
  ...
  <Msg ovf:msgid="info.os">Betriebssystem</Msg>
  ...
</Strings>
```

在顶部的 Strings（字符串）字段中，使用 xml:lang 属性来规定具体外部消息文件的本地化。外部消息文件

Error! Unknown document property name.

## 开放虚拟化格式白皮书

中含有本地化消息的 `Msg` 元素，可供 OVF 描述文件使用。

另一种使用本地化资源的方法是根据当前的位置引用外部文件。这种方法可以用于，例如根据位置显示许可协议文本。许可协议文本按照位置包含在文本文件中。下面这个示例中展示了如何通过引用外部文本文件来显示本地化的许可协议：

```
<Envelope xml:lang="en-US">
  <References>
    <File ovf:id="license-en-US" ovf:href="license-en-US.txt"/>
    <File ovf:id="license-de-DE" ovf:href="license-de-DE.txt"/>
  </References>
  ...
  <VirtualSystem ovf:id="...">
    <EulaSection>
      <Info>Licensing agreement</Info>
      <License ovf:msgid="license">Unused</License>
    </EulaSection>
    ...
  </VirtualSystem>
  ...
  <Strings xml:lang="en-US">
    <Msg ovf:msgid="license" ovf:fileRef="license-en-US">Invalid license</Msg>
  </Strings>
  <Strings xml:lang="de-DE">
    <Msg ovf:msgid="license" ovf:fileRef="license-de-DE">Ihre Lizenz ist nicht
    gültig</Msg>
  </Strings>
</Envelope>
```

`License`（许可）元素中包括有一个 `ovf:msgid` 属性。在 `Strings`（字符串）字段中，不同位置的 `ovf:msgid` 将链接到使用 `ovf:fileRef` 属性引用的一个链接上。`ovf:fileRef` 属性在 OVF 描述文件的 `References`（引用）字段中也带有一个相对应的条目。`References`（引用）字段中的条目可以用于处理含有许可协议文本的外部文件。

### 5.3 可扩展性

OVF 规范允许通过多种方式在 OVF 描述文件中加入自定义元数据：

- 可以将新的字段元素定义为 `Section`（字段）置换群的一部分，在 OVF 架构允许出现字段的情况下使用；

Error! Unknown document property name.

- OVF 架构使用开放内容模型，其中所有现有类型都可以在加入元素之后进行扩展。在 OVF 架构中，使用命名空间为“##other”的 xs:any 属性来宣告扩展点；
- OVF 架构允许在现有类型中加入新的属性。

OVF 规范的设计目标在于确保向前和向后兼容。对于向前兼容性而言，这就意味着使用后期规范（或者自定义扩展）规定功能的 OVF 描述文件，必须能够被下列 OVF 用户理解：i) 根据之前版本规范编制的 OVF；或者 ii) 不了解具体扩展内容的用户。必须能够让 OVF 用户可靠、可预期、按照用户友好的方式，决定接受还是不接受包含有扩展项的 OVF 包。

### 5.3.1 置换群

OVF 支持开放内容模型，允许在其中加入额外的字段，也允许使用新内容扩展现有的字段。在扩展之时，通过布尔值 ovf:required（要求）属性来规定：为确保正确的行为，相关信息是强制要求的还是可选的。

还可以通过在 ovf:Section（字段）置换群中定义新的成员，在 OVF 描述文件中插入其他字段。这就意味着新的字段可以扩展 Section（字段）元素。新的字段可以用于定义与 OVF 规范中规定的现有字段不相关的元数据。新的 Section（字段）元素中带有有一个<Info>（信息）元素，用于在部署功能不能理解字段的情况下，向用户提供关于扩展字段的信息。

下面这个示例中，描述了加入新字段<ns:BuildInformationSection>的情况，该字段使用 ns 命名空间。命名空间引用自母元素，即<Envelope>（包络）元素：<Envelope xmlns="http://schemas.dmtf.org/ovf/envelope/2" xmlns:ns="http://acme.org/ovf/extension/ns">。根据 ovf:Section substitutionGroup 要求，在新的字段中含有一个<Info>（信息）元素。其中 BuildNumber（构建编号）、BuildDate（构建日期）、以及 BuildSystem（构建系统）都是新元素。这些元素都引用的命名空间架构中进行定义。ovf:required（要求）属性的取值为“false”，这就表示在不执行这些字段的情况下，部署功能将会发出警告，但不会导致部署失败。

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

添加上面那个示例中所述的字段的 XSD 架构如下所述：

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns=http://acme.org/ovf/extension/ns
  xmlns:ovf=http://schemas.dmtf.org/ovf/envelope/2
  xmlns:xs=http://www.w3.org/2001/XMLSchema
  targetNamespace=http://acme.org/ovf/extension/ns
```



```
</xs:complexType>
</xs:schema>
```

在上述架构中，在 `ovf:Section` 字段中规定了 `BuildInformationSection`（构建信息字段）置换群。`BuildInformationSection`（构建信息字段）置换群的类型为 `BuildInformationSection_Type`（构建信息字段\_类型）。`BuildInformationSection_Type`（构建信息字段\_类型）中将 `ovf:Section_Type`（字段\_类型）规定视为基本类型，并通过 `BuildNumber`（构建编号）、`BuildDate`（构建日期）、以及 `BuildSystem`（构建系统）元素来扩展 `ovf:Section_Type`（字段\_类型）。

### 5.3.2 元素

可以在字段末尾，在现有字段中加入新元素。其中 `Envelope`（包络）、`VirtualSystem`（虚拟系统）、`VirtualSystemCollection`（虚拟系统集合）、`Content`（内容）、以及 `Strings`（字符串）字段不支持在字段末尾添加新元素。使用的命名空间需要引用自母元素，必须与 `OVF` 命名空间不同。可以使用其他元素来扩展 `OVF` 描述文件中特定字段中所包含的信息。

对现有字段进行扩展的示例如下所述：

```
<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>
```

在上面这个示例中，显示了在 `Annotation`（注释）字段中加入新元素的情况。此元素扩展了 `Annotation`（注释）字段，纳入了关于描述文件 `Author`（作者）相关的信息。此新元素属于 `ns` 命名空间。

### 5.3.3 属性

第三种扩展 `OVF` 描述文件的方法是在现有元素中加入新的自定义属性。这些属性可以用于扩展现有元素所提供的信息。

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

在上面这个示例中，显示的是在 `Network`（网络）元素中加入 `desiredCapacity`（要求能力）的属性。此新的属性将在 `ns` 命名空间中进行定义。

更多关于 `OVF` 文档扩展的详细信息，请参阅附件 E 相关内容。



## 5.4 符合性

在 OVF 规范中，规定了 OVF 描述文件的三个符合性层级，其中一级是最高的符合性层级。

- OVF 描述文件中只包括 OVF 规范中规定的元数据，即没有任何自定义扩展性：  
符合性层级：一级。
- OVF 描述文件中包括有自定义扩展元数据，但所有扩展项都为可选扩展：  
符合性层级：二级。
- OVF 描述文件中包括有自定义扩展元数据，且至少有一项扩展项是强制要求扩展：  
符合性层级：三级。

使用三级符合性层级会限制 OVF 的可移植性，这就意味着 OVF 包可能不能在除了支持自定义扩展项的平台之外的其他任何虚拟平台上部署。

## 5.5 虚拟硬件描述

下文中所述的硬件描述是相对较为概括的。具体而言，其中规定了需要虚拟磁盘和网络适配器。其中没有规定应该采用哪些具体的硬件，例如是采用 SCSI 还是 IDE 磁盘，是采用 E1000 还是 Vlanice 网卡等。更加具体地讲，可以假设如果规范文件只是概述性的，则虚拟应用必须能够发现存在的设备、并加载相关磁盘。在这种情况下，可以假设虚拟应用创建者应该使用广泛的磁盘开发设备，并对虚拟应用中的相关虚拟硬件进行测试，确保其能够正常工作。

如果在虚拟平台上部署的 OVF 包没有提供客户端软件需要、且包括在虚拟应用中的部分硬件设备，以及/或者设备分类，则安装将会失败。另外，如果用户不能解决问题，则还可能存在安装以及/或者引导失败的风险。在出现这种情况时，还会导致用户技术支持请求增加、客户不满意度增加的风险。更加具有限制性、更加详细的虚拟硬件规范可以降低不正确执行的风险（因为列出了要求采用的具体设备），但此规范只对虚拟应用能够安装以及/或者引导的系统数量做出了限制。

由于独立软件供货商选择虚拟应用模式的主要原因是考虑到安装到便捷性、稳固性和可预期性，因此虚拟应用开发商应该开放具有更多而不是更少广泛性的设备，除非相关虚拟应用具有非常具体的硬件要求。毕竟从本质上看，虚拟应用的可移植性主要取决于虚拟机可以使用的客户端软件，以及客户端软件支持的虚拟硬件范围。

在理想情况下，虚拟应用供货商将会创建带有适用于供货商目标虚拟平台的虚拟硬件的虚拟机。尽管如此，现在的很多虚拟平台供货商并不会发布独立于虚拟应用供货商/创建者的驱动。相反，为了进一步简化虚拟硬件-虚拟应用的界面管理，OVF 模型支持简单的安装模式。在此模式中，在安装完毕之后，即可以对每台虚拟机进行引导，以便允许进行具体虚拟平台的本地化/自定义。此模式允许虚拟机在首次引导时检测虚拟平台，并向客户端提供并安装正确的设备驱动，包括任何平台具体的驱动（例如，在首次引导时通过客户端的软盘或者光盘载入驱动）。除此之外，对于 Windows VM 而言，只需要使用命名等属性进行重新安

## 开放虚拟化格式白皮书

Error! Unknown document property name.

装和自定义，通过重启技术可以在用户不干预的情况下，实现镜像的命名和自定义。

下面这个示例中描述了多种虚拟硬件概况，适用于描述文件中规定的各种虚拟平台：

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

此类型的概况允许供货商对硬件描述进行自定义，以便支持不同的虚拟平台和功能。可以在其支持的特定的硬件字段中，选择具体的虚拟平台，同时假设 OVF 部署功能会选择本地平台上最新或者最实用的功能组。

下面这个示例中描述了可以定义的具体虚拟硬件的类型。可以通过 `rasd:ResourceSubType` 定义多种方案，并使用空格进行分隔。之后，部署功能可以选择实例化的具体虚拟硬件类型。

如果在 OVF 包中规定了多种资源子类型，则部署功能将会选择适宜的取值，来填充受管理环境的 `CIM_ResourceAllocationSettingData`（CIM\_资源配置设置数据）分级。如果存在多个适宜的取值，则部署功能将会从中任意选取取值。如果部署功能没有找到适宜的取值，则部署将会失败。

```
<Item>
  <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
  <rasd:InstanceID>1000</rasd:InstanceID>
  <rasd:ResourceSubType>LsiLogic BusLogic</rasd:ResourceSubType>
  <rasd:ResourceType>6</rasd:ResourceType>
```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
</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>
```

## 5.6 描述文件示例

在规范中提供了下面这些示例，作为 OVF 描述文件的完整示例。这些示例都通过了 XML 验证。

附件 A 中描述了单一虚拟系统的 OVF 描述文件；

附件 B 中描述了多套虚拟系统的 OVF 描述文件；

附件 C 中描述了 LAMP 堆栈中含有多个引用程序的单一虚拟系统的 OVF 描述文件；

附件 D 中描述了 LAMP 堆栈中含有多个引用程序、且带有两套虚拟系统的多重虚拟系统的 OVF 描述文件；

## 6. OVF 包部署

### 6.1 部署

部署是指将 OVF 包中的虚拟机转换为目标虚拟平台能够理解的运行时格式，同时进行适宜的资源配置，并获得正确的虚拟硬件支持。在部署过程中，平台会对 OVF 的完整性进行验证，确保 OVF 包在转移过程中没有被修改；同时还会检查其与本地虚拟硬件之间的兼容性。另外，在部署过程中，还会想虚拟机分配资源，对虚拟机进行配置，使其能够适合于目标虚拟平台的具体环境。此流程中包括与虚拟机连接的网络（实体网络和虚拟网络）资源的分配和配置。；向虚拟机分配存储器资源，包括虚拟硬盘、瞬态数据组、与集群或者网络化存储器连接等；对 CPU 和内存资源进行配置；对应用程序层面的属性进行自定义等。OVF 不支持在处理器架构或者硬件平台之间进行客户端软件转化。通过部署可以将带有与 OVF 描述文件中规定的要求兼容的一台或多台虚拟机实例化，同时包括一组包含有 OVF 包中所固定内容的虚拟磁盘。

OVF 包的部署体验主要取决于在其上部署的虚拟平台。可以通过命令行、脚本、或者图形部署向导进行部署。典型的部署工具将会显示或者提示下列信息：

- 显示关于 OVF 包的信息（来自于 ProductSection（产品字段）），并询问用户是否接受许可协议，或者执行无人安装功能；
- 验证虚拟硬件是否与 OVF 规定兼容；

Error! Unknown document property name.

- 询问用户虚拟机的储存位置，以及实体网络是否与 OVF 包中规定的逻辑网络连接；
- 要求用户输入在 ProductSection（产品字段）中进行配置的属性的具体取值。

在配置完成之后，预计虚拟机可以成功启动，并获得（使用诸如 DHCP 等标准流程）局域网内的有效身份。用于提示具体 IP 网络配置的属性和其他取值将会随部署环境的不同而不同。在虚拟应用首次启动之后，可能需要对设备内部的软件进行必要的额外配置，可以通过虚拟应用本身提供的管理界面进行配置，例如网页界面。

## 6.2 OVF 环境描述文件

OVF 环境描述文件是一个 XML 文档，其中描述了安装在虚拟磁盘中的软件的元数据信息。OVF 规范中规定了软件部署过程中常用的子元素，例如虚拟硬件、磁盘、网络、资源要求、以及自定义参数等。描述文件具有可扩展性，因此可以在后续加入新的信息。

虚拟应用通常需要进行自定义，以确保其能够在其部署的具体环境中发挥正常功能。OVF 环境能够为虚拟平台提供一种标准、可扩展的方式，以便将部署配置告知客户端软件。

OVF 环境是一个 XML 文档，其中含有客户端软件的部署时间自定义信息。可以在 XML 文档中提供的信息的示例包括：

- 操作系统层级的配置信息，例如主机名称、IP 地址、子网络、网关等；
- 应用程序层级的配置信息，例如活跃的目录服务器 DNS 名称、数据库、以及其他外部服务等。

在部署过程中，可以使用 ProductSection（产品字段）元数据，对 OVF 描述文件中规定的一组属性进行配置。在部署过程中，可以使用向导式的界面，让客户输入相关配置的取值。

例如，OVF 环境允许客户端软件自动执行不同层级服务之间的网络设置，网页服务器可以在不需要用户手动干预的情况下，自动配置数据库服务器的 IP 地址。

规定标准的 OVF 环境可能会带来一定的挑战，因为不存在标准的跨供货商的半虚拟化设备，能够支持虚拟机中的客户端软件和潜在的虚拟平台之间的通信。OVF 规范采用的方法是将 OVF 环境定义分为下列两个部分：

- 一个标准协议，其中规定了具有哪些可用信息，以及其可用格式；
- 一个传输属性，规定了如何获取信息的方式。

规范要求所有实施都支持 ISO 传输，以确保动态生成的 ISO 镜像中的客户端软件能使用 OVF 环境（XML 文件）。

## 6.3 部署过程中的资源配置选项

OVF 包能够为虚拟应用纳入资源配置选项。这就可以确保 OVF 的用户能够便捷地进行初始设置，不需要根据预计用途做出单独的资源决策。在 Description（描述）和 Label（标签）元素中，提供了人类可读的

Error! Unknown document property name.

## 开放虚拟化格式白皮书

资源配置列表，例如：

- 软件评估设置；
- 10-100 人工作组设置；
- 100-1000 人工作组设置；
- 大企业工作组设置。

部署功能会提示用户选择配置。上文所述的配置列表可以用作合适的初始资源配置。

配置列表示例如下所述：

```
<DeploymentOptionSection>
  <Configuration ovf:id="eval">
    <Label>Software Evaluation</Label>
    <Description>Software evaluation setup</Description>
  </Configuration>
  <Configuration ovf:id="small" ovf:default="yes">
    <Label>Small</Label>
    <Description>10-100 person workgroup setup</Description>
  </Configuration>
  <Configuration ovf:id="medium">
    <Label>Medium</Label>
    <Description>100-1000 person workgroup setup</Description>
  </Configuration>
  <Configuration ovf:id="large">
    <Label>Large</Label>
    <Description>Large enterprise workgroup setup</Description>
  </Configuration>
</DeploymentOptionSection>
```

OVF 描述文件中的下列片段描述了资源要求中资源配置选项的使用情况。在此示例中，如果用户选择“eval”，则所使用的资源配置将位于 `Item ovf:configuration="eval"` 中。

资源要求示例：

```
<ResourceAllocationSection>
  <Info>Defines reservations for CPU and memory</Info>
  <Item>
    ... default configuration ...
  </Item>
```

Error! Unknown document property name.

```
<Item ovf:configuration="eval">
    ... replaces the default configuration if the "eval" configuration if selected ...
</Item>
</ResourceAllocationSection>
```

在下面这个片段中，OVF 描述文件中描述了用于 VirtualHardwareSection（虚拟硬件字段）的配置选项。在此示例中，如果用户选择的是“large”，则所使用的资源配置将位于 Item ovf:configuration="large" 中。

VirtualHardwareSection（虚拟硬件字段）示例：

```
<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:Limit>1100</rasd:Reservation>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
</Item>
...
<Item ovf:configuration="large">
    <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
    <rasd:ElementName>1 CPU and 800 MHz reservation</rasd:ElementName>
    <rasd:InstanceID>1</rasd:InstanceID>
    <rasd:Reservation>800</rasd:Reservation>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
</Item>
</VirtualHardwareSection>
```

### 6.4 属性元素部署过程中的产品自定义

OVF 描述文件中包括有对客户端软件描述，其中包括关于在 OVF 环境中进行自定义的信息。此信息可以通过使用 OVF 描述文件中的 ProductSection（产品字段）中的 Property（属性）元素提供。

每个 Property（属性）元素都包括下列五个属性：ovf:key、ovf:type、ovf:qualifiers、ovf:value、以及 ovf:userConfigurable。

ovf:key 属性是一个独特的识别符，用于识别 Property（属性）元素；

Error! Unknown document property name.

开放虚拟化格式白皮书

ovf:type 属性用于说明 Property（属性）元素中所包含的取值的类型；

ovf:qualifiers 属性用于提供 Property（属性）元素的取值；

ovf:value 属性用于为 Property（属性）元素赋值。

ovf:userConfigurable 属性用于确定所提供的取值是在部署时可以变更的默认值，还是不可以变更的取值。

Property（属性）元素的示例如下所述：

```

<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>
  <Property ovf:key="Gateway" ovf:type="string" ovf:value="192.168.0.1"
    ovf:userConfigurable="false" >
    <Description>Gateway address to be used</Description>
  </Property>
  <Property ovf:key=" SoftwareResourceType" ovf:type="uint16"
    ovf:qualifiers="uint16,uint16,uint16,uint16,uint16,uint16,uint16,uint16,uint16,
      uint16,"
    ovf:value="Unknown", "Buffer", "Queue", "Protocol Endpoint", "Remote Interface",
      "Pool", "Cache", "File", "Database",
    ovf:userConfigurable="false" >
    <Description>Value Map example based on SoftwareResourceType property in
      CIM_SoftwareResource class</Description>
  </Property>
</ProductSection>

```

在 CIM\_SoftwareResource (CIM\_软件资源) 分级中, 有一个名为 SoftwareResourceType (软件资源类型) 的属性, 下面列出了其在 CIM 架构中的片段:

```

[Description (
  "The type of the software resource. Although the behavior "
  "of the different software resource types is modeled "
  "similarly, different names for resources transferring "
  "data over time or/and space have been established. "

```



Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
"SoftwareResourceType conveys their original, most common "  
"name. \n"  
ValueMap { "0", "2", "3", "4", "5", "6", "7", "8", "9",  
"10..32767", "32768..65535" },  
Values { "Unknown", "Buffer", "Queue", "Protocol Endpoint",  
"Remote Interface", "Pool", "Cache", "File", "Database",  
"DMTF Reserved", "Vendor Reserved" }]  
uint16 SoftwareResourceType;
```

ovf.type 与 ValueMap（取值映射）中的系数相一致，因此为“uint16”。

附件 D 中包括了对一个复杂的多层应用程序进行自定义的示例。

## 7. 可移植性

OVF 是一种能够提高虚拟应用以及其相关的虚拟机的可移植性的技术。OVF 包中包括有创建虚拟机的配方，可以被虚拟平台简明扼要地进行解释。在安装虚拟应用的过程中，通过打包的元数据能够提供稳固、用户友好的体验。更加具体地讲，管理基础设施可以使用元数据来可信地确定 OVF 中的虚拟机是否可以安装，同时可以用于引导适宜的转化和本地化处理，确保其在安装的具体执行环境中可以使用。

存在很多超出 OVF 格式规范控制范围的因素，即使在严格按照本规范要求实施的情况下，这些因素也会影响打包的虚拟机的可移植性。也就是说，将虚拟机打包为 OVF 包并不能保证在所有管理程序中具有普遍的可移植性或可安装性。下面是一些可能限制可移植性的因素：

- OVF 中的虚拟机中可能含有其格式不能被试图安装的管理程序理解的虚拟磁盘。尽管可以合理预期，绝大多数管理程序都能够采用任何一种主要的虚拟磁盘格式来导入和导出虚拟机，但还是可能会出现 OVF 支持、但具体的管理程序不支持的较新格式。
- 安装的客户端软件可能不支持管理程序内的虚拟硬件。例如，Xen 管理程序在默认情况下就不会向客户端软件提供虚拟软盘设备。因此，可以预期的是，对于需要与软盘控制器进行互动的客户端虚拟机，将无法正确执行虚拟机程序。
- 安装的客户端软件可能不支持相关的 CPU 架构。例如，客户端软件可能只能执行具体处理器模式的 CPU 操作，或者需要具体的浮点支持，或者含有具体 CPU 供货商特有的操作码。
- 虚拟平台可能无法理解 OVF 描述文件中要求的功能。例如，可能虚拟平台不支持组合服务。由于 OVF 标准与具体的虚拟产品无关，在任一点上，OVF 都可能无法获得在 OVF 规范发行之前定义的虚拟平台的支持。

OVF 的可移植性可以分为如下所述的等级：

- **可移植性等级 1:** 可以在多组虚拟硬件上运行。例如，虚拟应用可以使用 Xen、Sun、Microsoft、以及 VMware 管理程序。对于三级兼容性而言，必须开发出支持多种管理程序的客户端软件。执行客户端软件的清洁安装和引导，在此过程中客户端软件能够发现硬件设备，并安装与虚拟平台互动所需要的驱动程序，这就是 OVF 三级可移植性的一个示例。Microsoft Windows®操作系统的“sysprep”级别可移植性是另一个示例。上述客户端软件实例可以在多种硬件平台上重新安装、重新命名、重新自定义，包括虚拟硬件。
- **可移植性等级 2:** 可以在具体的一组虚拟硬件上运行。这种情况通常适用于缺乏安装的客户端软件的驱动支持导致的。
- **可移植性等级 3:** 只能在特定的虚拟产品以及/或者 CPU 架构以及/或者虚拟硬件选项上运行。这通常是由于 OVF 中含有废除的虚拟机、或者含有启动后虚拟机的快照导致的，包括真实或模拟设备的 CPU 的当前运行时状态。上述状态将 OVF 绑定到了非常具体的虚拟和硬件平台上。

对于在组织机构内部使用而言，2 级或 3 级兼容性可能就足够了，因为 OVF 包可以在受控环境中发布，通过对硬件和虚拟平台做出非常具体的采购决策，可以确保 OVF 潜在功能组的一致性。通过简单地导出虚拟机即可以创建 3 级或 2 级可移植性的 OVF（绑定到特定的一组硬件上）；尽管如此，可以非常容易地扩展隐喻，使其能够支持 1 级可移植性，例如通过使用 Windows 的“sysprep”功能。

对于由独立软件供货商独立创建和发布的商业虚拟应用而言，要求具有一级可移植性。实际上，一级可移植性可以确保相关设备能够在最广泛的客户基础上得到应用，包括用于评估和用于生产。应该使用工具套装创建特定的“已知合格”一级可移植性虚拟应用，以便能够在多种虚拟平台上进行发布和安装；如果相关虚拟应用预计只会在范围交债的虚拟硬件组合环境中使用，例如在企业内部的特定开发组中使用，则也可以创建 2 级可移植性平台。

OVF 虚拟硬件描述可以设计为支持一级到三级的可移植性。对于一级可移植性而言，可以在其中包括对硬件要求的较为概况的描述，或者具体规定多种替代虚拟硬件描述。虚拟应用供货商应该对虚拟硬件描述的灵活性或限制性具有完全的控制权限。范围较窄的规范可以用于限制虚拟应用只在已知合格的虚拟硬件上运行，但这会在一定程度上限制其可移植性。较为广泛的规定能够让虚拟应用在较广泛的虚拟硬件上允许。这能够确保客户具有最佳的客户体验，也是虚拟应用概念成果的一个主要要求。

## 附件 A（供参考）：单一虚拟系统示例

描述文件中绝大部分都是样本文件。开始应该在描述文件中加入对一组文件的描述。在本示例所述的情况下，只有一个文件（vmdisk1.vmdk）。之后，应该对虚拟应用使用的一组虚拟磁盘和网络进行描述。每种文件、磁盘和网络资源都应该分配一个独特的识别符。这些识别符应该放在不同的命名空间中，但最好的做法是使用不同的名称。

示例 OVF 的内容是单一虚拟机，内容中含有五个字段。

下面列出了典型单一虚拟机设备的完整 OVF 描述文件：

```
<?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 services are 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="adminEmail" ovf:type="string">
      <Description>Email address of administrator</Description>
    </Property>
    <Property ovf:key="appIp" 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>
    </Item>
  </VirtualHardwareSection>
</VirtualSystem>

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<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>
```

## 附件 B（供参考）：多层次 PetStore 示例

Pet Store OVF 描述文件中描述了多个先进的 OVF 概念：

- 多虚拟机包：使用 VirtualMachineCollection（虚拟机集合）实体子类型；
- 组合服务结构：使用嵌套的 VirtualMachineCollection（虚拟机集合）实体子类型；
- 传播用户自定义的部署配置；
- 可以使用 OVF 环境，实现服务部署时间的自定义；
- 使用虚拟磁盘链来减少下载量；
- 嵌套了 ProductSection（产品字段）元素，以便提供关于在具体虚拟机上安装的软件的相关信息。

本示例中所述的服务称为 Pet Store，其中包括前端网页服务器和数据库。数据库服务器本身是一个复杂的多层服务器，包括两台虚拟机，以确保容错率。

### B.1 架构和打包

Pet Store OVF 包中包括三套虚拟系统（WebTier、DB1 和 DB2），以及两个虚拟系统集合（Pet Store 和 DBTier）。图 B-1 中显示了 OVF 包的结构及其属性，以及虚拟机的启动顺序。

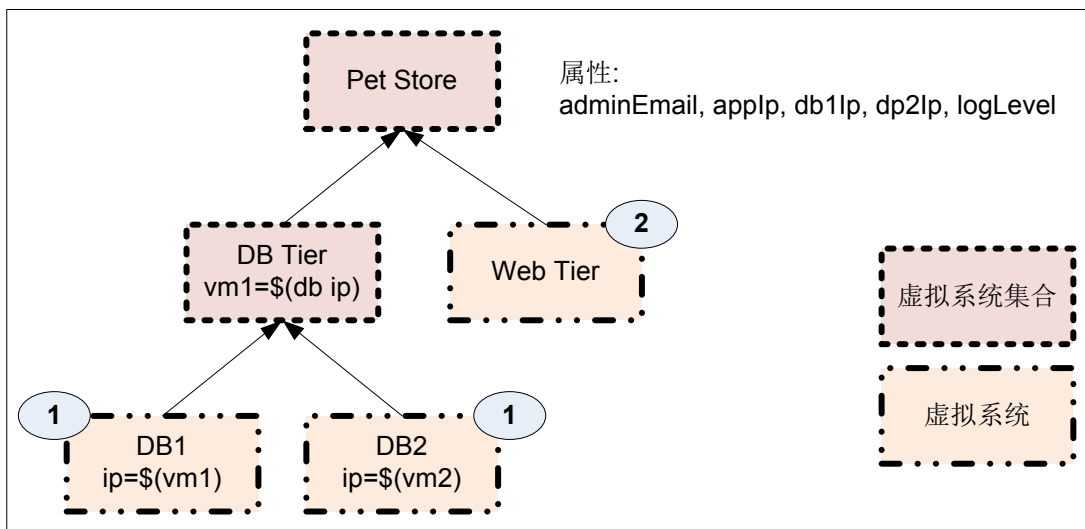


图 B-1: Pet Store OVF 包

完整的 OVF 描述文件将在本附件的末尾提供。OVF 的属性用途和磁盘布置将在本附件后续条款中详细介绍。

### B.2 属性

Pet Store 服务中包括五个用户可配置的属性。这些属性都是服务的重要控制参数，需要进行配置，以确保其能够在部署环境中正常启动。这些属性将以 OVF 环境文件的形式传输给客户端软件。客户端软件在启

Error! Unknown document property name.

## 开放虚拟化格式白皮书

动时能够读取 OVF 环境，提取这些属性的取值，将其应用到软件配置中。因此，OVF 描述文件能够反映客户端软件处理的属性。

对于此具体服务而言，有两种不同的软件配置：一种是网页层面、另一种是数据库层面。每种软件配置所支持的属性将在下文的表格中进行介绍。

表 B-1 中列出了网页基客户端软件的属性：

**表 B-1：网页层面配置**

属性	描述
<i>applp</i>	网页服务器的 IP 地址。
<i>dbIp</i>	与其连接的数据库服务器的 IP 地址。
<i>adminEmail</i>	技术支持的电子邮件地址。
<i>logLevel</i>	登录等级。

所有属性都应该在母元素 `VirtualSystemCollection`（虚拟系统集合）中进行定义，这些属性可供该集合下辖的 `VirtualSystem`（虚拟系统）或者 `VirtualSystemCollection`（虚拟系统集合）子元素使用。因此，OVF 描述文件不需要含有针对每台虚拟机的明确的 `ProductSection`（产品字段），正如 WebVM 所证实的一样。

表 B-2 中描述了数据库客户端软件的属性：

**表 B-2：数据库层面配置**

属性	描述
<i>Ip</i>	虚拟机的 IP 地址。
<i>primaryAtBoot</i>	选择在引导是该实例为主引导设备还是从引导设备。
<i>ip2</i>	作为热备份或者主设备的第二个数据库的 IP 地址。
<i>log</i>	登录水平（简称为登录）。

集群数据库可以组织为虚拟系统集合，其中应该带有一组配置属性：`vm1`、`vm2` 和 `log`。通过这种组织方式，可以将数据库实施与 OVF 包中的软件的其余部分分离，从而能够允许虚拟应用（客户端软件+虚拟机配置）能够容易地进行组合，促进回用。

数据库软件是一种标准软件包，供货商会选择“`com.mydb.db`”作为所有属性的独特名称。在 OVF 描述文件中可以看到此字符串，此字符串被纳入到 `ProductSection`（产品字段）元素中的 `ovf:class` 属性中。

在 OVF 描述文件的实体层级中，使用\${<name>}属性语法来将取值从外层传输给内部的节点。通过此机制可以将不同的部件连接起来，不需要提前规定命名公约，也不需要修改客户端软件。只有在 VirtualSystemCollection（虚拟系统集合）母元素中定义的属性才能够被子元素直接使用。因此，Petstore 虚拟系统定义的属性不能被 DB1 虚拟系统使用。这就可以确保 VirtualSystemCollection（虚拟系统集合）元素可以被包含在 VirtualSystemCollection（虚拟系统集合）母元素中，并进行良好描述，这可以让软件具有可组合性和容易使用性。

OVF 描述文件使用固定的非用户可配置的属性来确定两台数据库虚拟机在引导过程中具有不同的功能，即使是通过完全相同的软件镜像进行引导。名为 com.mydb.db.primaryAtBoot 的属性具有一个固定的、非用户可配置的取值，但对于两种镜像而言取值不同。软件在引导时会检查此取值，并根据此取值自定义其操作。

### B.3 磁盘布置

Pet Store OVF 包具有共享磁盘、以及对德尔塔磁盘架构进行编码的能力，能够减少文件体积，从而缩短下载时间。在此具体示例中，我们只介绍了两种不同的镜像（数据库和网页），如果我们进一步假设其构建在相同的基础操作系统的顶层，则我们可以按照图 B-2 中所述的方式对此配置进行编码。

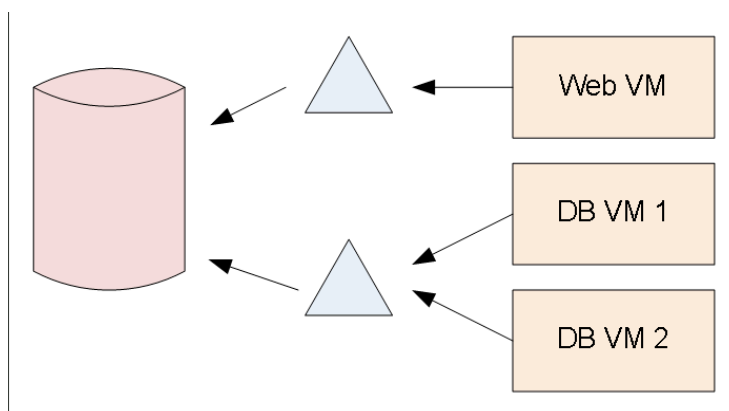


图 B-2: Pet Store 虚拟磁盘布置

因此，尽管 OVF 包中含有三台不同的虚拟机，但其总下载数据量也非常小。实际上，只需要下载一台完整的虚拟机和两个相对较小的德尔塔虚拟磁盘即可。

在部署系统中，虚拟磁盘的物理布置与 OVF 包中规定的磁盘结构无关。OVF 包只描述虚拟磁盘的容量和内容（即：磁盘上需要具有的字节）。另外，其中还规定每台虚拟机都应该具有独立的硬盘。因此，在安装上述 OVF 包时，虚拟平台可以将其视为具有三个独立硬盘的三台迅疾，也可以选择复制上述结构，也可以采用其他形式，只需要确保每台虚拟机都能够访问含有初始引导时所需内容的磁盘，同时一台虚拟机的写入操作不会影响其他虚拟机。

### B.4 Pet Store OVF 描述文件

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



Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
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 services
        are 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>
  </DeploymentOptionSection>
</ovf:Envelope>
```

```

<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 multitier 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>
  <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>

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<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>

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
</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>
  <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>

```



Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<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>
```

## B.5 完整 OVF 环境

在下面这个示例中，列出了 WebTier 和 DB1 虚拟机可以访问的 OVF 环境（DB2 是与 DB1 完全相同的虚拟机，因此在此忽略）。

WebTier 虚拟机的 OVF 环境：

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

```

</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 虚拟机的 OVF 环境:

```

<?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"/>

```

Error! Unknown document property name.

开放虚拟化格式白皮书

```
<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>
```

## 附件 C（供参考）：单一虚拟系统 LAMP 堆栈示例

在本附件中，我们将提供两个示例，用于说明 LAMP 虚拟应用的 OVF 描述文件的具体情况。我们同时提供了单虚拟机 LAMP 虚拟应用和多虚拟机 LAMP 虚拟应用的描述文件。LAMP 是使用 Linux 系统构建的服务，其中包括 Apache 网页服务器、MySQL 数据块、以及 PHP 网页开发软件包。

此示例中描述了如何使用 ProductSection（产品字段）元素来同时规定操作系统和应用程序层面上的部署参数。例如，在部署在特定的环境中之后，这些参数可以用于优化服务的性能。这些描述文件都是完整的文件，但尽可能以简明的方式提供。例如，其中没有包括 EULA（最终用户许可协议）字段。

### C.1 部署时自定义

在 OVF 包部署阶段内可以提供自定义参数。自定义参数将在 OVF 描述文件中做出规定，并利用 OVF 环境提供给客户端软件。部署时自定义可以加入到虚拟机层面的参数中，起重工包括虚拟开关的连接性、以及物理存储器的位置。

对于 LAMP 基虚拟应用而言，部署时自定义包括服务的 IP 地址和端口编号，诸如网关和子网络等网络信息，另外还有其他参数，以确保可以对特定部署进行优化。部署者可以访问的属性随供货商和服务的不同而不同。在示例描述文件中，可以针对四种不同的 LAMP 部件使用下文表 3 中列出的一组属性。

表 C-1: LAMP 配置

产品	属性	描述
Linux	<i>Hostname</i> （主机名称）	应用程序的网络参数，包括 IP 地址。
	<i>IP</i>	
	<i>Subnet</i> （子网络）	
	<i>Gateway</i> （网关）	
	<i>Dns</i>	
	netCoreRemeMax	用于优化 IP 堆栈传输速度的参数。
	netCoreWemeMax	
Apache	httpPort	网页服务器端口编号。
	HttpsPort	
	startThreads	用于优化网页服务器的参数。
	minSpareThreads	

产品	属性	描述
	maxSpareThreads	
	maxClients	
MySQL	queryCacheSize	用于优化数据库性能的参数。
	maxConnections	
	waitTimeout	
PHP	sessionTimeout	用于自定义 PHP 引擎的参数，包括会话超时和会话数量方面的设置。
	concurrentSessions	
	memoryLimit	

使用斜体字表示的属性是用户可以进行配置的属性。其他属性是指具有合理的默认值，因此用户不需要为其提供取值的属性。

每种软件产品的自定义参数都包含在各自的产品字段中。例如，对于 Apache 网页服务器而言，使用的是下列字段：

```
<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">
```

Error! Unknown document property name.

```

        <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 are served. </Description>
    </Property>
</ProductSection>

```

ovf:class="org.apache.httpd"属性规定了这些属性的前缀。因此，Apache 数据库在 OVF 环境中预计可以访问下列属性：

```

<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>

```

## C.2 简单 LAMP OVF 描述文件

带有 LAMP 堆栈的单一虚拟机虚拟应用的完整 OVF 描述文件如下所述：

```

<?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 -->

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<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 is 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 are
    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>
  </Item>

```

Error! Unknown document property name.  
开放虚拟化格式白皮书

```
<rasd:ResourceType>17</rasd:ResourceType>  
</Item>  
</VirtualHardwareSection>  
</VirtualSystem>  
</Envelope>
```

## 附件 D（供参考）：多重虚拟系统 LAMP 堆栈示例

此示例是多虚拟机 LAMP 虚拟应用的 OVF 描述文件示例。LAMP 是使用 Linux 系统构建的服务，其中包括 Apache 网页服务器、MySQL 数据块、以及 PHP 网页开发软件包。

### D.1 双层 LAMP OVF 描述文件

在双层 LAMP 堆栈中，应用程序层（Linux、Apache、PHP）和数据库层（Linux、MySQL 服务器）是在不同的虚拟机上运行的，主要目的是确保更高的可扩展性。

OVF 格式确保了服务实施过程对于用户而言是透明的。更加具体地讲，在用户安装单虚拟机或者双层 LAMP 设备时，部署体验是基本类似的。唯一的区别在于用户需要支持两个 IP 地址，以及两个 DNS 主机名称。

与单虚拟机描述文件相比进行了下列改动：

- 所有用户可配置的参数都放置在 VirtualSystemCollection（虚拟系统集合）实体中。与单虚拟机示例相比，Apache、MySQL 和 PHP 的 ProductSection（产品字段）元素保持不变；
- 两台虚拟机上的 Linux 软件需要使用不同的 IP 地址和主机名称进行配置，尽管其余绝大多数参数都可以共享。在 VirtualSystemCollection（虚拟系统集合）中需要加入一个新的 ProductSection（产品字段）元素，以便提示用户，同时应该使用 \${property} 表达为每个 VirtualSystem（虚拟系统）实体分配取值；
- 使用磁盘链来确保下载量与单虚拟机设备基本相当。由于 Linux 软件是储存在共享的磁盘上，因此只需要下载一份 Linux 副本即可以有效安装。

完整的 OVF 描述文件示例如下所述：

```
<?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"/>
  </References>
</Envelope>
```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<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 is 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 ovf:class="org.mylamp">
    <Info>Product information for the service</Info>
    <Product>My Lamp Service</Product>
    <Version>1.0</Version>
    <FullVersion>1.0.0</FullVersion>
  </ProductSection>
  <ProductSection ovf:class="org.linuxdist">
    <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 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"

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
        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 are
        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>

```



Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
</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>
    <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>

```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
<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-db</rasd:HostResource>
  <rasd:InstanceID>5</rasd:InstanceID>
  <rasd:Parent>4</rasd:Parent>
  <rasd:ResourceType>17</rasd:ResourceType>
</Item>
</VirtualHardwareSection>
</VirtualSystem>
</VirtualSystemCollection>
</Envelope>
```

## 附件 E（供参考）：可扩展性示例

OVF 规范允许通过多种方式在 OVF 描述文件中加入自定义元数据：

- 可以将新的字段元素定义为 **Section**（字段）置换群的一部分，在 OVF 架构允许出现字段的情况下使用；
- OVF 架构使用开放内容模型，其中所有现有类型都可以在加入元素之后进行扩展。在 OVF 架构中，使用命名空间为“**##other**”的 **xs:any** 属性来宣告扩展点；
- OVF 架构允许在现有类型中加入新的属性。

自定义元数据不允许使用 OVF XML 命名空间。在自定义元素中，使用一个布尔值 **ovf:required** 属性来规定元素中的相关信息是确保正常操作所必须的，还是可选的。

OVF 架构中的开放内容模型只允许在末尾处扩展现有类型。使用 XML 1.0 架构将无法便捷地支持更具灵活性的开放内容模型，这是因为独特粒子规则，以及需要在架构中的很多地方加入 **xs:any** 宣告。XML 1.1 架构草案标准中带有更加灵活的开放内容机制，可以使用 **xs:openContent mode="interleave"** 进行宣告。

### E.1 自定义架构

在自定义的 XML 架构中，规定了如下所述的两种扩展类型。第一种宣告规定了 OVF Section（字段）置换群的自定义成员；第二种宣告规定了简单自定义类型。

```
<?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>
</xs:schema>
```

Error! Unknown document property name.

## 开放虚拟化格式白皮书

```
        </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 >
```

## E.2 带有自定义扩展项的描述文件

完整的 OVF 描述文件使用如下所述的自定义架构。描述文件可以根据 OVF 架构和自定义架构进行验证，但没有包括扩展项的示例，这主要是为了确保描述文件示例的简洁性。

描述文件中含有所有三种扩展类型：自定义的 OVF Section（字段）元素、一个扩展点的自定义元素、以及一个自定义属性。

```
<?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 metadata 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 环境 XML 架构中含有与此 OVF 包络 XML 架构相匹配的扩展机制，因此 OVF 环境文件也可以进行类似的扩展。

## 附件 F（供参考）：修订记录

版本	日期	描述
1.0.0	2009 年 2 月 17 日	
1.0.1	2011 年 10 月 20 日	发布 DMTF 勘误版本。
2.0.0	2014 年 4 月 24 日	发布 DMTF 标准，将其定性为“供参考规范”。