

# View 集成指南

VMware Horizon 7 7.0



vmware®

您可以从 VMware 网站下载最新的技术文档:

<https://docs.vmware.com/cn/>。

如果您对本文档有任何意见或建议, 请将反馈信息发送至:

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

**威睿信息技术(中国)有限公司**  
北京办公室  
北京市  
朝阳区新源南路 8 号  
启皓北京东塔 8 层 801  
[www.vmware.com/cn](http://www.vmware.com/cn)

上海办公室  
上海市  
淮海中路 333 号  
瑞安大厦 804-809 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

广州办公室  
广州市  
天河路 385 号  
太古汇一座 3502 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

版权所有 © 2019 VMware, Inc. 保留所有权利。 [版权和商标信息](#)

# 目录

## View 集成指南 6

### 1 View 集成简介 7

View 组件 7

View 的集成接口 8

### 2 将 View 与事件数据库相集成 9

事件数据库表和结构定义 9

连接代理事件 11

Horizon Agent 事件 16

View Administrator 事件 17

事件消息属性 25

示例数据库查询和视图 26

### 3 使用 View PowerCLI 29

View PowerCLI 使用入门 29

启动加载了 View PowerCLI 的 PowerShell 控制台 30

从远程系统使用 View PowerCLI cmdlet 30

显示 View PowerCLI 的帮助 31

检查 View PowerCLI cmdlet 错误 31

通过管道传输并指定相同类型的对象 31

转义 vCenter Server 路径名称中的字符 32

View Administrator、PowerCLI Cmdlet 和 View 命令行界面比较 32

View PowerCLI cmdlet 参考 37

View PowerCLI cmdlet 参数 38

使用 View PowerCLI cmdlet 的示例 42

管理 View 连接服务器实例 42

在 View 中管理 vCenter Server 实例 43

管理桌面池 43

创建和更新自动置备的桌面池 44

创建和更新链接克隆桌面池 44

创建和更新手动置备的桌面池 45

创建和更新手动未受管桌面池 45

显示有关用户和组的信息 46

管理桌面授权 46

管理远程会话 46

管理虚拟机 47

显示有关物理机的信息	47
更新虚拟机所有权	47
显示事件报告	48
显示和更新全局设置	48
显示和添加许可证密钥	48
使用 View PowerCLI 执行高级任务的示例	49
确定 View 连接服务器是否正在运行	49
调整自动和链接克隆池的大小	50
确定 vSphere 清单对象的路径	51
确定 vSphere 数据存储对象的路径	52
添加和移除数据存储	52
向桌面池分配多个网络标签	54
网络标签配置文件格式	54
网络标签配置文件示例	55
获取并导出网卡和网络标签信息	56
验证并编辑网络标签配置文件	57
部署使用多个网络标签的桌面池	58
显示池的网络标签分配	60
显示虚拟机的网络标签分配	61
显示虚拟机的 vCenter Server 网络标签分配	61
禁用自动网络标签分配	62
<b>4 自定义 LDAP 数据</b>	<b>63</b>
LDAP 配置数据简介	63
修改 LDAP 配置数据	63
导出 LDAP 配置数据	64
在 LDIF 配置文件中定义桌面池	64
使用 vdmimport 命令导入 LDAP 配置数据	67
使用 LDIFDE 命令导入 LDAP 配置数据	68
<b>5 将 View 与 Microsoft SCOM 相集成</b>	<b>69</b>
设置 SCOM 集成	69
为 View 连接服务器组分配名称	69
View 管理包	70
在 SCOM 服务器上导入 View 管理包	70
在 View 连接服务器主机或安全服务器上启用代理程序	71
在 Operations Manager 控制台中运行发现脚本	71
View 连接服务器和安全服务器受管对象	72
View 对象类和关系	72
在 Operations Manager 控制台中监控 View	74
用于 View 的视图和监视器	74

- 显示发现的受管 View 对象 76
- 显示性能信息 76
- 显示 View 连接服务器组的警示 76
- 关闭警示 76
- 重新启动 View 组件服务 77
- 从连接监控中排除域 77

## 6 使用 WMI 查看 PCoIP 会话统计信息 78

- 使用 PCoIP 会话统计信息 78
- 常规 PCoIP 会话统计信息 79
- PCoIP 音频统计信息 80
- PCoIP 图像处理统计信息 80
- PCoIP 网络统计信息 81
- PCoIP USB 统计信息 83
- 使用 PowerShell cmdlet 查看 PCoIP 统计信息的示例 83

## 7 使用启动会话脚本设置桌面策略 85

- 获取启动会话脚本的输入数据 85
- 使用启动会话脚本的最佳做法 85
- 准备 View 桌面以使用启动会话脚本 86
  - 启用 VMware View 脚本主机服务 86
  - 为启动会话脚本添加 Windows 注册表项 87
- 示例启动会话脚本 88

# View 集成指南

《View 集成指南》文档介绍如何将 View™ 软件与第三方软件集成，例如 Windows PowerShell、商业智能报告引擎和 Microsoft System Center Operations Manager (SCOM)。

## 目标读者

本文档面向任何想要通过自定义或集成软件来与 View 一起使用的用户。本文档的目标读者为熟悉虚拟机技术和数据中心操作且具有丰富经验的 Windows 或 Linux 系统管理员。

# View 集成简介

使用 View，系统管理员可以置备桌面，并控制用户对这些桌面的访问。客户端软件可将用户连接到在 VMware vSphere™ 中运行的虚拟机，或连接到在您的网络环境中运行的物理系统。此外，View 管理员可以配置远程桌面服务 (Remote Desktop Service, RDS) 主机，以向客户端设备提供 View 桌面和应用程序会话。

本章讨论了以下主题：

- View 组件
- View 的集成接口

## View 组件

您可以结合使用 View 与 VMware vCenter Server，以通过在 VMware ESX® 或 VMware ESXi™ 主机上运行的虚拟机创建桌面，并将这些桌面部署到最终用户。您也可以在 RDS 主机上安装 View，以将桌面和应用程序部署到最终用户。View 可利用您现有的 Active Directory 基础架构对用户进行身份验证和管理。

创建桌面或应用程序后，授权的最终用户可以使用基于 Web 或本地安装的客户端软件安全地连接到集中式虚拟机、后端物理系统或 RDS 主机。

View 包括以下主要组件。

<b>View 连接服务器</b>	一种充当客户端连接代理的软件服务，负责执行身份验证并将入站用户请求引导到相应的虚拟机、物理系统或 RDS 主机。
<b>Horizon Agent</b>	一种安装在所有客户机虚拟机、物理系统或 RDS 主机上以允许通过 View 进行管理的软件服务。Horizon Agent 提供连接监控、虚拟打印、USB 支持和单点登录等功能。
<b>Horizon Client</b>	一种可与 View 连接服务器进行通信以使用户能够连接到其桌面的软件应用程序。
<b>View Administrator</b>	一种 Web 应用程序，允许 View 管理员配置 View 连接服务器、部署桌面和应用程序池、管理计算机、控制用户身份验证、启动和检查系统事件，以及执行分析活动。

---

**注** View Administrator 在 Horizon 7 中名为 Horizon Administrator。

---

<b>vCenter Server</b>	一种可充当网络上连接的 ESX/ESXi 主机的中心管理员的服务器。vCenter Server 实例提供了配置、置备和管理数据中心中的虚拟机的中央点。
<b>View Composer</b>	一种安装在 vCenter Server 实例上的软件服务，允许 View 从一个集中式基础映像快速部署多个链接克隆桌面。

## View 的集成接口

您可以使用多个接口将 View 与外部应用程序相集成。

<b>事件数据库</b>	您可以将 View 配置为将事件记录到 Microsoft SQL Server 或 Oracle 数据库。然后，您可以使用商业智能报告引擎来访问和分析此数据库。
<b>View PowerCLI</b>	您可以使用 PowerShell 接口对 View 组件执行多种管理任务。
<b>轻型目录访问协议 (Lightweight Directory Access Protocol, LDAP)</b>	您可以将 LDAP 配置数据导入到 View（或者从 View 导出）。您可以创建更新此配置数据的脚本，而无需直接访问 View Administrator。
<b>Microsoft System Center Operations Manager (SCOM)</b>	您可以从 SCOM 控制台监控 View 组件的操作。
<b>Windows Management Instrumentation (WMI)</b>	您可以查看 PCoIP 会话的性能统计信息。



## 将 View 与事件数据库相集成

您可以将 View 配置为将事件记录到 Microsoft SQL Server 或 Oracle 数据库。View 记录各类事件，如最终用户操作、管理员操作、报告系统故障和错误的警示及统计抽样。

最终用户操作包括记录和启动桌面和应用程序会话。管理员操作包括添加授权及创建桌面和应用程序池。例如，记录 24 个小时内的最大用户数量即是一种统计抽样。

您可以使用 Crystal Reports、IBM Cognos、MicroStrategy 9 和 Oracle Enterprise Performance Management System 等业务智能报告引擎来访问和分析事件数据库。

本章讨论了以下主题：

- [事件数据库表和结构定义](#)
- [连接代理事件](#)
- [Horizon Agent 事件](#)
- [View Administrator 事件](#)
- [事件消息属性](#)
- [示例数据库查询和视图](#)

### 事件数据库表和结构定义

View 使用数据库表实施事件数据库。事件数据库会在这些表的名称前面附加您在设置数据库时定义的前缀。

#### 事件数据库表

下表显示了用来在 View 中实施事件数据库的数据库表。

**表 2-1. 事件数据库表**

表名称	说明
event	最新事件的元数据和搜索优化数据。
event_data	最新事件的数据值。
event_data_historical	所有事件的数据值。
event_historical	所有事件的元数据和搜索优化数据。

**View** 将有关事件的详细信息记录到所有数据库表中。自写入事件记录经过特定的时间段之后，**View** 将从 **event** 和 **event\_data** 表中删除该记录。您可以使用 **View Administrator** 配置数据库在 **event** 和 **event\_data** 表中保留记录的时间长度。

**重要事项** **View** 不会限制 **event\_historical** 和 **event\_data\_historical** 表的增长。您必须为这些表实施空间管理策略。

作为唯一主键，**EventID** 标识 **View** 在 **event** 和 **event\_historical** 表中记录的每个事件。**View** 将每个事件的数据值记录在 **event\_data** 和 **event\_data\_historical** 表中。您可以通过使用“**EventID**”列联接 **event** 和 **event\_data** 表或联接 **event\_historical** 和 **event\_data\_historical** 表，来获取某个事件的完整信息。

**Event** 和 **event\_historical** 表中的“**EventType**”、“**Severity**”和“**Time**”列标识事件的类型和严重性以及事件的发生时间。

有关设置事件数据库的信息，请参阅《**View** 安装指南》文档。

**注** 如果您在事件数据库未运行时重新启动 **View** 连接服务器实例，事件可能会丢失。关于避免此问题的解决方案，请参阅 <http://kb.vmware.com/kb/1021461>。

## 事件数据库结构定义

下表显示了 **event** 和 **event\_historical** 数据库表的结构定义。

**表 2-2. Event 和 event\_historical 表的结构定义**

列名称	Oracle 数据类型	SQL Server 数据类型	说明
Acknowledged	SMALLINT	tinyint	<b>View</b> 是否已确认事件。 ■ 0 = false ■ 1 = true
DesktopId	NVARCHAR2(512)	nvarchar(512)	关联池的桌面 ID。
EventID	整数	int	事件的唯一主键。
EventType	NVARCHAR2(512)	nvarchar(512)	与消息目录中的项目对应的事件名称。例如， <b>BROKER_USERLOGGEDIN</b> 。
FolderPath	NVARCHAR2(512)	nvarchar(512)	包含已关联对象的文件夹的完整路径。
GroupId	NVARCHAR2(512)	nvarchar(512)	<b>Active Directory</b> 中已关联组的 <b>SID</b> 。
LUNId	NVARCHAR2(512)	nvarchar(512)	存储已关联对象的 <b>LUN</b> 的 <b>ID</b> 。
MachineId	NVARCHAR2(512)	nvarchar(512)	关联的物理机或虚拟机的 <b>ID</b> 。
Module	NVARCHAR2(512)	nvarchar(512)	引发事件的 <b>View</b> 组件。例如， <b>Admin</b> 、 <b>Broker</b> 、 <b>Tunnel</b> 、 <b>Framework</b> 、 <b>Client</b> 或 <b>Agent</b> 。
ModuleAndEventText	NVARCHAR2(512)	nvarchar(512)	包含替换属性参数的值的事件消息。
Node	NVARCHAR2(512)	nvarchar(512)	虚拟设备节点的名称。
严重性	NVARCHAR2(512)	nvarchar(512)	严重性级别。例如， <b>INFO</b> 、 <b>WARNING</b> 、 <b>ERROR</b> 、 <b>AUDIT_SUCCESS</b> 、 <b>AUDIT_FAIL</b> 。

列名称	Oracle 数据类型	SQL Server 数据类型	说明
源	NVARCHAR2(512)	nvarchar(512)	事件源的标识符。
ThinAppId	NVARCHAR2(512)	nvarchar(512)	关联的 ThinApp™ 对象的 ID。
时间	TIMESTAMP	datetime	事件的发生时间，以新纪元时间（1970 年 1 月 1 日）为准。
UserDiskPathId	NVARCHAR2(512)	nvarchar(512)	用户磁盘的 ID。
UserSID	NVARCHAR2(512)	nvarchar(512)	Active Directory 中已关联用户的 SID。

下表显示了 event\_data 和 event\_data\_historical 数据库表的结构定义。

**表 2-3. event\_data 和 event\_data\_historical 表的结构定义**

列名称	Oracle 数据类型	SQL Server 数据类型	说明
BooleanValue	SMALLINT	tinyint	布尔属性的值。 ■ 0 = false ■ 1 = true
EventID	整数	int	事件的唯一主键。
IntValue	整数	int	整数属性的值。
名称	NVARCHAR2(512)	nvarchar(512)	属性名称（例如，UserDisplayName）。
StrValue	NVARCHAR2(512)	nvarchar(512)	字符串属性的值。对于其他类型的属性，此列包含该数据类型的字符串解释。
TimeValue	TIMESTAMP	datetime	日期和时间属性的值。
类型	SMALLINT	tinyint	属性的数据类型。 ■ 0 = StrValue ■ 1 = IntValue ■ 2 = TimeValue ■ 3 = BooleanValue

## 连接代理事件

连接代理事件报告与 View 连接服务器相关的信息，如桌面和应用程序会话、用户身份验证失败和置备错误。

BROKER\_DAILY\_MAX\_DESKTOP\_SESSIONS 事件报告 24 小时内并发桌面会话的最大数量。如果用户同时运行多个桌面会话，则将单独计数每个桌面会话。

BROKER\_DAILY\_MAX\_APP\_USERS 事件报告 24 小时内并发应用程序用户的最大数量。如果用户同时运行多个应用程序，则仅将该用户计入一次。由于每五分钟执行一次采样，短期会话可能不会计入计数中。

BROKER\_VC\_DISABLED 和 BROKER\_VC\_ENABLED 事件报告 View 用来跟踪 vCenter Server 实例的 vCenter 驱动程序的状态。

BROKER\_VC\_STATUS\_\* 事件报告 vCenter Server 实例的状态。

下表列出了 View 连接服务器的所有事件类型。

**表 2-4. 连接代理事件**

事件类型	严重性	ModuleAndEventText
BROKER_AGENT_OFFLINE	BROKER_AGENT_OFFLINE 警告	在计算机\${MachineName} 上运行的代理未响应查询，将其标记为脱机
BROKER_AGENT_ONLINE	警告	在计算机\${MachineName} 上运行的代理正在重新响应，但未发送启动消息
BROKER_APPLICATION_LAUNCH_FAILURE	错误	无法为用户 \${UserDisplayName} 从池 \${PoolId} 中启动: 代理在处理请求时遇到错误，请联系支持人员以寻求帮助
BROKER_APPLICATION_MISSING	警告	池 \${PoolId} 中的 \${MachineName} 上至少安装了 \${ApplicationMissingCount} 个应用程序，包括 \${ApplicationExecutable}
BROKER_APPLICATION_NOT_ENTITLED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${PoolId} 中启动: 用户未获得此池的授权
BROKER_APPLICATION_PROTOCOL_NOT_SUPPORTED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${PoolId} 中启动: 请求的协议 \${ProtocolId} 不受支持
BROKER_APPLICATION_REQUEST	信息	用户 \${UserDisplayName} 已请求应用程序 \${ApplicationId}
BROKER_APPLICATION_SESSION_REQUEST	信息	用户 \${UserDisplayName} 已从池 \${PoolId} 请求应用程序会话
BROKER_DAILY_MAX_DESKTOP_SESSIONS	信息	\$(Time): 在过去的 24 小时内，并行桌面会话的最大数量为 \${UserCount} 个
BROKER_DAILY_MAX_APP_USERS	信息	\$(Time): 在过去的 24 小时内，具有并行应用程序会话的用户的最大数量为 \${UserCount} 个
BROKER_DESKTOP_LAUNCH_FAILURE	错误	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 代理在处理请求时遇到错误，请联系支持人员以寻求帮助
BROKER_DESKTOP_NOT_ENTITLED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 用户未获得此池的授权
BROKER_DESKTOP_PROTOCOL_NOT_SUPPORTED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 请求的协议 \${ProtocolId} 不受支持
BROKER_DESKTOP_REQUEST	信息	用户 \${UserDisplayName} 已请求池 \${DesktopId}
BROKER_EVENT_HANDLING_STARTED	信息	代理 \${BrokerName} 已开始处理事件
BROKER_EVENT_HANDLING_STOPPED	信息	\$(BrokerName) 已停止处理事件
BROKER_MACHINE_ALLOCATED	信息	用户 \${UserDisplayName} 已请求池 \${DesktopId}，已分配计算机 \${MachineName}
BROKER_MACHINE_ASSIGNED_UNAVAILABLE	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 分配的计算机 \${MachineName} 不可用
BROKER_MACHINE_CANNOT_CONNECT	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 使用 \${ProtocolId} 连接计算机 \${MachineName} 失败

事件类型	严重性	ModuleAndEventText
BROKER_MACHINE_CONFIGURED_VIDEO_SETTINGS	信息	在池 \${DesktopId} 中为虚拟机\${MachineName} 成功配置了视频设置
BROKER_MACHINE_NOT_READY	警告	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 计算机 \${MachineName} 未准备好接受连接
BROKER_MACHINE_OPERATION_DELETED	信息	计算机\${MachineName} 已被删除
BROKER_MACHINE_PROTOCOL_NOT_SUPPORTED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 计算机 \${MachineName} 不支持协议 \${ProtocolId}
BROKER_MACHINE_PROTOCOL_UNAVAILABLE	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 计算机 \${MachineName} 未报告协议 \${ProtocolId} 已就绪
BROKER_MACHINE_REJECTED_SESSION	警告	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 计算机\${MachineName} 已拒绝启动会话请求
BROKER_MACHINE_SESSION_TIMEDOUT	警告	用户 \${UserDisplayName} 的会话已超时
BROKER_MULTIPLE_DESKTOPS_FOR_KIOSK_USER	警告	用户 \${UserDisplayName} 获得多个桌面池的授权
BROKER_POOL_CANNOT_ASSIGN	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 没有可将用户分配到的计算机
BROKER_POOL_COMANAGER_REQUIRED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 协议 \${ProtocolId} 不提供共同管理功能
BROKER_POOL_EMPTY	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 桌面池为空
BROKER_POOL_NO_MACHINE_ASSIGNED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 未给此用户分配计算机
BROKER_POOL_NO_RESPONSES	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 桌面池中的计算机均无响应
BROKER_POOL_OVERLOADED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 所有响应计算机当前都在使用中
BROKER_POOL_POLICY_VIOLATION	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 此桌面池不支持联机会话
BROKER_POOL_PROTOCOL_NOT_SUPPORTED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 没有计算机可以支持协议 \${ProtocolId}
BROKER_POOL_PROTOCOL_UNAVAILABLE	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 没有报告协议 \${ProtocolId} 已就绪的计算机
BROKER_POOL_TUNNEL_NOT_SUPPORTED	AUDIT_FAIL	无法为用户 \${UserDisplayName} 从池 \${DesktopId} 中启动: 协议 \${ProtocolId} 不支持安全加密链路
BROKER_PROVISIONING_ERROR_CONFIG_CLEARED	信息	以前报告的配置问题未在池 \${DesktopId} 中出现
BROKER_PROVISIONING_ERROR_CONFIG_SET	错误	由于存在配置问题, 池 \${DesktopId} 中出现置备错误
BROKER_PROVISIONING_ERROR_DISK_CLEARED	信息	以前报告的磁盘问题未在池 \${DesktopId} 中出现

事件类型	严重性	ModuleAndEventText
BROKER_PROVISIONING_ERROR_DISK_LC_RESERVATION_CLEARED	信息	以前报告的因可用磁盘空间已预留给链接克隆而出现的错误未在池 \${DesktopId} 中出现
BROKER_PROVISIONING_ERROR_DISK_LC_RESERVATION_SET	错误	由于可用磁盘空间已预留给链接克隆，池 \${DesktopId} 出现置备错误
BROKER_PROVISIONING_ERROR_DISK_SET	警告	由于存在磁盘问题，池 \${DesktopId} 中出现置备错误
BROKER_PROVISIONING_ERROR_LICENCE_CLEARED	信息	以前报告的许可问题未在池 \${DesktopId} 中出现
BROKER_PROVISIONING_ERROR_LICENCE_SET	错误	由于存在许可问题，池 \${DesktopId} 出现置备错误
BROKER_PROVISIONING_ERROR_NETWORKING_CLEARED	信息	以前报告的 Horizon Agent 存在的网络问题未在池 \${DesktopId} 中出现
BROKER_PROVISIONING_ERROR_NETWORKING_SET	错误	由于 Horizon Agent 存在网络问题，池 \${DesktopId} 出现置备错误
BROKER_PROVISIONING_ERROR_RESOURCE_CLEARED	信息	以前报告的资源问题未在池 \${DesktopId} 中出现
BROKER_PROVISIONING_ERROR_RESOURCE_SET	错误	由于存在资源问题，池 \${DesktopId} 出现置备错误
BROKER_PROVISIONING_ERROR_TIMEOUT_CUSTOMIZATION_CLEARED	信息	以前报告的自定义过程中出现的超时问题未在池 \${DesktopId} 中出现
BROKER_PROVISIONING_ERROR_TIMEOUT_CUSTOMIZATION_SET	错误	由于自定义过程中出现超时，池 \${DesktopId} 出现置备错误
BROKER_PROVISIONING_ERROR_VM_CLONING	错误	计算机\${MachineName} 出现置备错误: 克隆计算机失败
BROKER_PROVISIONING_ERROR_VM_CUSTOMIZATION_ERROR	错误	计算机\${MachineName} 出现置备错误: 自定义计算机失败
BROKER_PROVISIONING_ERROR_VM_CUSTOMIZATION_NETWORKING	错误	计算机 \${MachineName} 出现置备错误: 因 Horizon Agent 和连接服务器之间不存在网络通信导致自定义错误
BROKER_PROVISIONING_ERROR_VM_CUSTOMIZATION_TIMEOUT	错误	计算机\${MachineName} 出现置备错误: 自定义操作超时
BROKER_PROVISIONING_SVI_ERROR_COMPOSER_AGENT_INIT_FAILED	错误	计算机\${MachineName} 出现置备错误: View Composer 代理初始化失败
BROKER_PROVISIONING_SVI_ERROR_RECONFIG_FAILED	错误	计算机\${MachineName} 出现置备错误: 重新配置操作失败
BROKER_PROVISIONING_SVI_ERROR_REFIT_FAILED	错误	计算机\${MachineName} 出现置备错误: 重新调整操作 \${SVIOperation} 失败
BROKER_PROVISIONING_SVI_ERROR_REMOVING_VM	错误	计算机\${MachineName} 出现置备错误: 无法从清单中移除计算机
BROKER_PROVISIONING_VERIFICATION_FAILED_USER_ASSIGNED	警告	计算机 \${MachineName} 的置备验证失败: 已将用户分配给池 \${DesktopId} 中的计算机

事件类型	严重性	ModuleAndEventText
BROKER_PROVISIONING_VERIFICATION_FAILED_USER_CANNOT_BE_ASSIGNED	警告	计算机 \${MachineName} 的置备验证失败: 由于池 \${DesktopId} 并非为持久池而无法分配用户
BROKER_PROVISIONING_VERIFICATION_FAILED_VMNAME_IN_USE	警告	计算机 \${MachineName} 的置备验证失败: 池 \${DesktopId} 中已存在名为 \${MachineName} 的计算机
BROKER_SECURITY_SERVER_ADD_FAILED	AUDIT_FAIL	添加安全服务器 \${SecurityServerId} 失败
BROKER_SECURITY_SERVER_ADD_FAILED_PASSWORD_EXPIRED	AUDIT_FAIL	添加安全服务器 \${SecurityServerId} 失败, 配对密码过期
BROKER_SECURITY_SERVER_ADD_FAILED_PASSWORD_INCORRECT	AUDIT_FAIL	添加安全服务器 \${SecurityServerId} 失败, 配对密码错误
BROKER_SECURITY_SERVER_ADD_FAILED_PASSWORD_NOT_SET	AUDIT_FAIL	添加安全服务器 \${SecurityServerId} 失败, 未设置配对密码
BROKER_SECURITY_SERVER_ADDED	AUDIT_SUCCESS	已添加安全服务器 \${SecurityServerId}
BROKER_SVI_ARCHIVE_UDD_FAILED	AUDIT_FAIL	未能将用户数据磁盘 \${UserDiskName} 归档到位置 \${SVIPath}
BROKER_SVI_ARCHIVE_UDD_SUCCEEDED	AUDIT_SUCCESS	已将用户数据磁盘 \${UserDiskName} 归档到位置 \${SVIPath}
BROKER_SVI_ATTACH_UDD_FAILED	AUDIT_FAIL	未能将用户数据磁盘 \${UserDiskName} 附加到虚拟机 \${SVIVMID}
BROKER_SVI_ATTACH_UDD_SUCCEEDED	AUDIT_SUCCESS	已将用户数据磁盘 \${UserDiskName} 附加到虚拟机 \${SVIVMID}
BROKER_SVI_DETACH_UDD_FAILED	AUDIT_FAIL	未能将用户数据磁盘 \${UserDiskName} 与虚拟机 \${SVIVMID} 分离
BROKER_SVI_DETACH_UDD_SUCCEEDED	AUDIT_SUCCESS	已将用户数据磁盘 \${UserDiskName} 与虚拟机 \${SVIVMID} 分离
BROKER_USER_AUTHFAILED_ACCOUNT_DISABLED	AUDIT_FAIL	用户 \${UserDisplayName} 由于帐户被禁用未能通过身份验证
BROKER_USER_AUTHFAILED_ACCOUNT_EXPIRED	AUDIT_FAIL	用户 \${UserDisplayName} 由于帐户已过期未能通过身份验证
BROKER_USER_AUTHFAILED_ACCOUNT_LOCKED_OUT	AUDIT_FAIL	用户 \${UserDisplayName} 由于帐户被锁定未能通过身份验证
BROKER_USER_AUTHFAILED_ACCOUNT_RESTRICTION	AUDIT_FAIL	用户 \${UserDisplayName} 由于帐户限制未能通过身份验证
BROKER_USER_AUTHFAILED_BAD_USER_PASSWORD	AUDIT_FAIL	用户 \${UserDisplayName} 由于用户名或密码错误未能通过身份验证
BROKER_USER_AUTHFAILED_GENERAL	AUDIT_FAIL	用户 \${UserDisplayName} 未能通过身份验证
BROKER_USER_AUTHFAILED_NO_LOGON_SERVERS	AUDIT_FAIL	用户 \${UserDisplayName} 由于没有登录服务器未能通过身份验证
BROKER_USER_AUTHFAILED_PASSWORD_EXPIRED	AUDIT_FAIL	用户 \${UserDisplayName} 由于密码已过期未能通过身份验证
BROKER_USER_AUTHFAILED_PASSWORD_MUST_CHANGE	AUDIT_FAIL	用户 \${UserDisplayName} 由于密码必须更改未能通过身份验证

事件类型	严重性	ModuleAndEventText
BROKER_USER_AUTHFAILED_SECUREID_ACCESS_DENIED	AUDIT_FAIL	用户 \${UserDisplayName} 的 SecurID 访问被拒绝
BROKER_USER_AUTHFAILED_SECUREID_NEWPIN_REJECTED	AUDIT_FAIL	用户 \${UserDisplayName} 的 SecurID 访问由于新 PIN 被拒绝而被拒绝
BROKER_USER_AUTHFAILED_SECUREID_WRONG_NEXTTOKEN	AUDIT_FAIL	用户 \${UserDisplayName} 的 SecurID 访问由于输入的下一个令牌错误而被拒绝
BROKER_USER_AUTHFAILED_SECUREID_WRONG_STATE	AUDIT_FAIL	用户 \${UserDisplayName} 的 SecurID 访问由于状态错误而被拒绝
BROKER_USER_AUTHFAILED_TIME_RESTRICTION	AUDIT_FAIL	用户 \${UserDisplayName} 由于时间限制未能通过身份验证
BROKER_USER_NOT_AUTHORIZED	AUDIT_FAIL	用户 \${UserDisplayName} 已经通过身份验证，但未获得执行此操作的授权
BROKER_USER_NOT_ENTITLED	AUDIT_FAIL	用户 \${UserDisplayName} 已经通过身份验证，但未获得任何池的授权
BROKER_USERCHANGEDPASSWORD	AUDIT_SUCCESS	\${UserDisplayName} 的密码已被用户更改
BROKER_USERLOGGEDIN	AUDIT_SUCCESS	用户 \${UserDisplayName} 已登录
BROKER_USERLOGGEDOUT	AUDIT_SUCCESS	用户 \${UserDisplayName} 已注销
BROKER_VC_DISABLED	信息	地址为 \${VCAddress} 的 vCenter 已被暂时禁用
BROKER_VC_ENABLED	信息	地址为 \${VCAddress} 的 vCenter 已被启用
BROKER_VC_STATUS_CHANGED_CANNOT_LOGIN	警告	无法登录地址为 \${VCAddress} 的 vCenter
BROKER_VC_STATUS_CHANGED_DOWN	信息	地址为 \${VCAddress} 的 vCenter 已关闭
BROKER_VC_STATUS_CHANGED_INVALID_CREDENTIALS	警告	地址为 \${VCAddress} 的 vCenter 具有无效凭据
BROKER_VC_STATUS_CHANGED_NOT_YET_CONNECTED	信息	尚未连接到地址为 \${VCAddress} 的 vCenter
BROKER_VC_STATUS_CHANGED_RECONNECTING	信息	正在重新连接地址为 \${VCAddress} 的 vCenter
BROKER_VC_STATUS_CHANGED_UNKNOWN	警告	地址为 \${VCAddress} 的 vCenter 的状态未知
BROKER_VC_STATUS_CHANGED_UP	信息	地址为 \${VCAddress} 的 vCenter 已启动

## Horizon Agent 事件

View Agent 事件报告与 Horizon Agent 相关的信息，如已登录特定计算机或从特定计算机断开连接的用户、特定计算机上的 Horizon Agent 是否已关闭，以及 Horizon Agent 是否已从特定计算机向 View 连接服务器发送启动消息。



表 2-5. Horizon Agent 事件

事件类型	严重性	ModuleAndEventText
AGENT_CONNECTED	信息	用户 \${UserDisplayName} 已在计算机\${MachineName} 上登录新会话
AGENT_DISCONNECTED	信息	用户 \${UserDisplayName} 已从计算机\${MachineName} 断开连接
AGENT_ENDED	信息	用户 \${UserDisplayName} 已注销计算机\${MachineName}
AGENT_PENDING	信息	在计算机\${MachineName} 上运行的代理已接受用户 \${UserDisplayName} 的已分配会话。
AGENT_PENDING_EXPIRED	警告	用户 \${UserDisplayName} 在计算机\${MachineName} 上的待处理会话已过期
AGENT_RECONFIGURED	信息	计算机\${MachineName} 已成功进行了重新配置
AGENT_RECONNECTED	信息	用户 \${UserDisplayName} 已重新连接到计算机\${MachineName}
AGENT_RESUME	信息	计算机\${MachineName} 上的代理已发送恢复消息
AGENT_SHUTDOWN	信息	在计算机\${MachineName} 上运行的代理已关闭，此计算机将不可用。
AGENT_STARTUP	信息	在计算机\${MachineName} 上运行的代理已联系连接服务器并发送了启动消息。
AGENT_SUSPEND	信息	计算机\${MachineName} 上的代理已发送挂起消息

## View Administrator 事件

View Administrator 事件报告有关用户在 View Administrator 中启动的操作的信息。

表 2-6. View Administrator 事件

EventType	严重性	ModuleAndEventText
ADMIN_ADD_DESKTOP_ENTITLEMENT	AUDIT_SUCCESS	\${EntitlementDisplay} 已通过 \${UserDisplayName} 获得了池 \${DesktopId} 的授权
ADMIN_ADD_LICENSE	AUDIT_SUCCESS	\${UserDisplayName} 已添加许可证
ADMIN_ADD_LICENSE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加许可证
ADMIN_ADD_PM	AUDIT_SUCCESS	\${UserDisplayName} 已将物理机\${MachineName} 添加到池 \${DesktopId}
ADMIN_ADD_PM_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将物理机\${MachineName} 添加到池 \${DesktopId}
ADMIN_ADD_THINAPP_ENTITLEMENT	AUDIT_SUCCESS	\${UserDisplayName} 已将应用程序 \${ThinAppDisplayName} 分配给桌面\${MachineName}
ADMIN_ADD_THINAPP_ENTITLEMENT_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加应用程序授权

EventType	严重性	ModuleAndEventText
ADMIN_ADD_THINAPP_POOL_ENTITLEMENT	AUDIT_SUCCESS	应用程序 \${ThinAppDisplayName} 已被 \${UserDisplayName} 分配给池 \${DesktopId}
ADMIN_ADMINISTRATOR_REMOVE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除管理员 \${AdminPermissionEntity} 的所有权限
ADMIN_ADMINISTRATOR_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除了管理员 \${AdminPermissionEntity} 的所有权限
ADMIN_CONNECTION_BROKER_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能更新连接代理 \${BrokerId}
ADMIN_CONNECTION_BROKER_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新连接代理 \${BrokerId}: (\${AttrChangeType}): \${AttrName} = \${AttrValue})
ADMIN_CONNECTION_SERVER_BACKUP_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能启动连接代理 \${BrokerId} 的备份
ADMIN_CONNECTION_SERVER_BACKUP_INITIATED	AUDIT_SUCCESS	\${UserDisplayName} 启动了连接代理 \${BrokerId} 的备份
ADMIN_CONNECTION_SERVER_DISABLE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能禁用连接代理 \${BrokerId}
ADMIN_CONNECTION_SERVER_DISABLED	AUDIT_SUCCESS	\${UserDisplayName} 正在禁用连接代理 \${BrokerId}
ADMIN_CONNECTION_SERVER_ENABLE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能启用连接代理 \${BrokerId}
ADMIN_CONNECTION_SERVER_ENABLED	AUDIT_SUCCESS	\${UserDisplayName} 正在启用连接代理 \${BrokerId}
ADMIN_DATABASE_CONFIGURATION_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加数据库配置
ADMIN_DATABASE_CONFIGURATION_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加数据库配置
ADMIN_DATABASE_CONFIGURATION_DELETE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能删除数据库配置
ADMIN_DATABASE_CONFIGURATION_DELETE_FAILED	AUDIT_SUCCESS	\${UserDisplayName} 已删除数据库配置
ADMIN_DATABASE_CONFIGURATION_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能更新数据库配置
ADMIN_DATABASE_CONFIGURATION_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新数据库配置
ADMIN_DEFAULT_DESKTOPPOOL_ASSIGN	AUDIT_SUCCESS	\${UserDisplayName} 已将默认桌面的池 \${DesktopId} 分配给 \${UserName}
ADMIN_DEFAULT_DESKTOPPOOL_ASSIGN_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将默认桌面的池 \${DesktopId} 分配给 \${UserName}
ADMIN_DEFAULT_DESKTOPPOOL_UNASSIGN	AUDIT_SUCCESS	\${UserDisplayName} 已移除对 \${UserName} 的默认桌面的池分配
ADMIN_DEFAULT_DESKTOPPOOL_UNASSIGN_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除对 \${UserName} 的默认桌面的池分配
ADMIN_DESKTOP_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加池 \${DesktopId}

EventType	严重性	ModuleAndEventText
ADMIN_DESKTOP_ASSIGN	AUDIT_SUCCESS	\${UserDisplayName} 已将桌面 \${MachineName} 分配给 \${UserName}
ADMIN_DESKTOP_ASSIGN_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将桌面 \${MachineName} 分配给 \${UserName}
ADMIN_DESKTOP_EDITED	AUDIT_SUCCESS	\${UserDisplayName} 已编辑池 \${DesktopId} (\${AttrChangeType}): \${AttrName} = \${AttrValue})
ADMIN_DESKTOP_MAINTENANCE_MODE_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将桌面 \${MachineName} 更新为 \${MaintenanceMode} 维护模式
ADMIN_DESKTOP_MAINTENANCE_MODE_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已将桌面 \${MachineName} 更新为 \${MaintenanceMode} 维护模式
ADMIN_DESKTOP_UNASSIGN	AUDIT_SUCCESS	\${UserDisplayName} 已移除桌面 \${MachineName} 的分配
ADMIN_DESKTOP_UNASSIGN_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除桌面 \${MachineName} 的分配
ADMIN_ENABLE_DESKTOP_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将池 \${DesktopId} 设置为 \${EnableStatus}
ADMIN_ENABLE_DESKTOP_SUCCEEDED	AUDIT_SUCCESS	\${UserDisplayName} 将池 \${DesktopId} 设置为 \${EnableStatus}
ADMIN_ENABLED_DESKTOP_PROVISION_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将池 \${DesktopId} 的置备设置为 \${EnableStatus}
ADMIN_ENABLED_DESKTOP_PROVISION_SUCCEEDED	AUDIT_SUCCESS	\${UserDisplayName} 将池 \${DesktopId} 的置备设置为 \${EnableStatus}
ADMIN_EVENT_CONFIGURATION_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能更新事件配置
ADMIN_EVENT_CONFIGURATION_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新全局配置
ADMIN_FOLDER_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加文件夹 \${AdminFolderName}
ADMIN_FOLDER_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加文件夹 \${AdminFolderName}
ADMIN_FOLDER_CHANGE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将对象 \${ObjectID}(type=\${ObjectType}) 更改为文件夹 \${AdminFolderName}
ADMIN_FOLDER_CHANGED	AUDIT_SUCCESS	\${UserDisplayName} 已将对象 \${ObjectID}(type=\${ObjectType}) 更改为文件夹 \${AdminFolderName}
ADMIN_FOLDER_DELETE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能删除文件夹 \${AdminFolderName}
ADMIN_FOLDER_DELETED	AUDIT_SUCCESS	\${UserDisplayName} 已删除文件夹 \${AdminFolderName}

EventType	严重性	ModuleAndEventText
ADMIN_GLOBAL_CONFIGURATION_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能更新全局配置
ADMIN_GLOBAL_CONFIGURATION_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新全局配置 (\$ {AttrChangeType}: \${AttrName} = \$ {AttrValue})
ADMIN_GLOBAL_POLICY_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能更新全局策略
ADMIN_GLOBAL_POLICY_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新全局策略 (\$ {AttrChangeType}: \${AttrName} = \$ {AttrValue})
ADMIN_PERFMON_CONFIGURATION_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能更新性能监视配置
ADMIN_PERFMON_CONFIGURATION_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新性能监视配置
ADMIN_PERMISSION_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能在文件夹 \$ {AdminFolderName} 上为具有 \$ {AdminRoleName} 角色的 \$ {AdminPermissionEntity} 添加权限
ADMIN_PERMISSION_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已在文件夹 \$ {AdminFolderName} 上为具有 \$ {AdminRoleName} 角色的 \$ {AdminPermissionEntity} 添加了权限
ADMIN_PERMISSION_REMOVE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能在文件夹 \$ {AdminFolderName} 上为具有 \$ {AdminRoleName} 角色的 \$ {AdminPermissionEntity} 移除权限
ADMIN_PERMISSION_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已在文件夹 \$ {AdminFolderName} 上为具有 \$ {AdminRoleName} 角色的 \$ {AdminPermissionEntity} 移除了权限
ADMIN_POOL_POLICY_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能更新池 \$ {DesktopId} 的策略
ADMIN_POOL_POLICY_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新池 \$ {DesktopId} 的策略 (\${AttrChangeType}: \$ {AttrName} = \${AttrValue})
ADMIN_REMOVE_DESKTOP_ENTITLEMENT	AUDIT_SUCCESS	\${EntitlementDisplay} 通过 \$ {UserDisplayName} 未获得池 \$ {DesktopId} 的授权
ADMIN_REMOVE_DESKTOP_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除池 \$ {DesktopId}
ADMIN_REMOVE_DESKTOP_SUCCEEDED	AUDIT_SUCCESS	\${UserDisplayName} 已移除池 \$ {DesktopId}
ADMIN_REMOVE_THINAPP_ENTITLEMENT	AUDIT_SUCCESS	\${UserDisplayName} 已取消从桌面 \$ {MachineName} 分配应用程序 \$ {ThinAppDisplayName}
ADMIN_REMOVE_THINAPP_ENTITLEMENT_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除应用程序授 权

EventType	严重性	ModuleAndEventText
ADMIN_REMOVE_THINAPP_POOL_ENTITLEMENT	AUDIT_SUCCESS	\${UserDisplayName} 已取消从池 \$ {DesktopId} 分配应用程序 \$ {ThinAppDisplayName}
ADMIN_RESET_THINAPP_STATE	AUDIT_SUCCESS	\${UserDisplayName} 已为桌面 \$ {DesktopDisplayName} 重置应用程序 \$ {ThinAppDisplayName} 的状态
ADMIN_RESET_THINAPP_STATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能重置 \$ {ThinAppDisplayName} 的应用程序状态
ADMIN_ROLE_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加具有 \$ {AdminPrivilegeName} 特权的角色 \$ {AdminRoleName}
ADMIN_ROLE_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加具有 \$ {AdminPrivilegeName} 特权的角色 \$ {AdminRoleName}
ADMIN_ROLE_PRIV_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将角色 \$ {AdminRoleName} 的权限更新为特权 \$ {AdminPrivilegeName}
ADMIN_ROLE_PRIV_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已将角色 \$ {AdminRoleName} 的权限更新为特权 \$ {AdminPrivilegeName}
ADMIN_ROLE_REMOVE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除角色 \$ {AdminRoleName}
ADMIN_ROLE_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除角色 \$ {AdminRoleName}
ADMIN_ROLE_RENAME_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将角色 \$ {AdminRoleName} 重命名为 \$ {AdminRoleNewName}
ADMIN_ROLE_RENAMED	AUDIT_SUCCESS	\${UserDisplayName} 已将角色 \$ {AdminRoleName} 重命名为 \$ {AdminRoleNewName}
ADMIN_SECURITY_SERVER_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加安全服务器 \${SecurityServerId}
ADMIN_SECURITY_SERVER_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加安全服务器 \$ {SecurityServerId}
ADMIN_SECURITY_SERVER_EDIT_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能编辑安全服务器 \${SecurityServerId}
ADMIN_SECURITY_SERVER_EDITED	AUDIT_SUCCESS	\${UserDisplayName} 已编辑安全服务器 \$ {SecurityServerId} (\${AttrChangeType}: \$ {AttrName} = \${AttrValue})
ADMIN_SECURITY_SERVER_REMOVE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除安全服务器 \${SecurityServerId}
ADMIN_SECURITY_SERVER_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除安全服务器 \$ {SecurityServerId}

EventType	严重性	ModuleAndEventText
ADMIN_SESSION_SENDMSG	AUDIT_SUCCESS	\${UserDisplayName} 已将消息 (\${SessionMessage}) 发送给会话 (用户 \${UserName}, 桌面 \${MachineName})
ADMIN_SESSION_SENDMSG_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能将消息 (\${SessionMessage}) 发送给会话 \${ObjectId}
ADMIN_SVI_ADD_DEPLOYMENT_GROUP_FAILED	AUDIT_FAIL	未能为 \${SVIParentVM} 添加部署组: \${SVISnapshot}
ADMIN_SVI_ADD_DEPLOYMENT_GROUP_SUCCEEDED	AUDIT_SUCCESS	已为 \${SVIParentVM} 添加了部署组 \${SVIDeploymentGroupId}: \${SVISnapshot}
ADMIN_SVI_ADD_UDD_FAILED	AUDIT_FAIL	未能添加用户数据磁盘 \${UserDiskName}
ADMIN_SVI_ADD_UDD_SUCCEEDED	AUDIT_SUCCESS	已添加用户数据磁盘 \${UserDiskName}
ADMIN_SVI_ADMIN_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加 SVI QuickPrep 域 \${SVIAdminFqdn} (\${SVIAdminName})
ADMIN_SVI_ADMIN_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除 SVI QuickPrep 域 (id=\${SVIAdminID})
ADMIN_SVI_ADMIN_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 已更新 SVI QuickPrep 域 \${SVIAdminFqdn} (\${SVIAdminName})
ADMIN_SVI_ATTACH_UDD_FAILED	AUDIT_FAIL	未能请求将用户数据磁盘 \${UserDiskName} 附加到虚拟机 \${SVIVMID}
ADMIN_SVI_ATTACH_UDD_SUCCEEDED	AUDIT_SUCCESS	已请求将用户数据磁盘 \${UserDiskName} 附加到虚拟机 \${SVIVMID}
ADMIN_SVI_DELETE_UDD_FAILED	AUDIT_FAIL	未能删除用户数据磁盘 \${UserDiskName}
ADMIN_SVI_DELETE_UDD_SUCCEEDED	AUDIT_SUCCESS	已删除用户数据磁盘 \${UserDiskName}
ADMIN_SVI_DETACH_UDD_FAILED	AUDIT_FAIL	未能请求将用户数据磁盘 \${UserDiskName} 与虚拟机 \${SVIVMID} 分离
ADMIN_SVI_DETACH_UDD_SUCCEEDED	AUDIT_SUCCESS	已请求将用户数据磁盘 \${UserDiskName} 与虚拟机 \${SVIVMID} 分离
ADMIN_SVI_REBALANCE_VM_FAILED	AUDIT_FAIL	未能重新平衡虚拟机 \${SVIVMID}
ADMIN_SVI_REBALANCE_VM_SUCCEEDED	AUDIT_SUCCESS	已重新平衡虚拟机 \${SVIVMID}
ADMIN_SVI_REFRESH_VM_FAILED	AUDIT_FAIL	未能刷新虚拟机 \${SVIVMID}
ADMIN_SVI_REFRESH_VM_SUCCEEDED	AUDIT_SUCCESS	已刷新虚拟机 \${SVIVMID}
ADMIN_SVI_RESYNC_VM_FAILED	AUDIT_FAIL	未能将虚拟机 \${SVIVMID} 与部署组 \${SVIDeploymentGroupId} 重新同步
ADMIN_SVI_RESYNC_VM_SUCCEEDED	AUDIT_SUCCESS	已将虚拟机 \${SVIVMID} 与部署组 \${SVIDeploymentGroupId} 重新同步

EventType	严重性	ModuleAndEventText
ADMIN_SVI_UPDATE_POOL_DEPLOYMENT_GROUP_FAILED	AUDIT_FAIL	未能将池 \${DesktopId} 更新为部署组 \${SVIDeploymentGroupID}
ADMIN_SVI_UPDATE_POOL_DEPLOYMENT_GROUP_SUCCEEDED	AUDIT_SUCCESS	已将池 \${DesktopId} 更新为部署组 \${SVIDeploymentGroupID}
ADMIN_SVI_UPDATE_UDD_FAILED	AUDIT_FAIL	未能更新用户数据磁盘 \${UserDiskName}
ADMIN_SVI_UPDATE_UDD_SUCCEEDED	AUDIT_SUCCESS	将用户数据磁盘 \${UserDiskName} 池设置为 \${DesktopId}，将用户设置为 \${UserName}
ADMIN_THINAPP_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加应用程序 \${ThinAppDisplayName}
ADMIN_THINAPP_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加应用程序 \${ThinAppDisplayName}
ADMIN_THINAPP_DESKTOP_AVAILABLE	AUDIT_SUCCESS	现在即可在桌面 \${DesktopDisplayName} 上使用应用程序 \${ThinAppDisplayName}
ADMIN_THINAPP_DESKTOP_REMOVED	AUDIT_SUCCESS	应用程序 \${ThinAppDisplayName} 已从桌面 \${DesktopDisplayName} 移除
ADMIN_THINAPP_EDITED	AUDIT_SUCCESS	\${UserDisplayName} 已编辑应用程序 \${ThinAppDisplayName}
ADMIN_THINAPP_FAILED_DESKTOP_DELIVERY	AUDIT_FAIL	未能将应用程序 \${ThinAppDisplayName} 提交给桌面 \${DesktopDisplayName}
ADMIN_THINAPP_FAILED_DESKTOP_REMOVAL	AUDIT_FAIL	未能从桌面 \${DesktopDisplayName} 移除应用程序 \${ThinAppDisplayName}
ADMIN_THINAPP_GROUP_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加应用程序模板 \${ThinAppGroupName}
ADMIN_THINAPP_GROUP_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已使用应用程序 \${ThinAppGroupApplications} 添加了应用程序模板 \${ThinAppGroupName}
ADMIN_THINAPP_GROUP_EDIT_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能编辑应用程序模板 \${ThinAppGroupName}
ADMIN_THINAPP_GROUP_EDITED	AUDIT_SUCCESS	\${UserDisplayName} 已使用应用程序 \${ThinAppGroupApplications} 编辑了应用程序模板 \${ThinAppGroupName}
ADMIN_THINAPP_GROUP_REMOVE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除应用程序模板 \${ThinAppGroupName}
ADMIN_THINAPP_GROUP_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除应用程序模板 \${ThinAppGroupName}
ADMIN_THINAPP_REMOVE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除应用程序 \${ThinAppDisplayName}
ADMIN_THINAPP_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除应用程序 \${ThinAppDisplayName}

EventType	严重性	ModuleAndEventText
ADMIN_THINAPP_REPO_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加存储库 \${ThinAppRepositoryName}，路径为 \${ThinAppRepositoryPath}
ADMIN_THINAPP_REPO_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加存储库 \${ThinAppRepositoryName}，路径为 \${ThinAppRepositoryPath}
ADMIN_THINAPP_REPO_EDIT_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能编辑存储库 \${ThinAppRepositoryName}，路径为 \${ThinAppRepositoryPath}
ADMIN_THINAPP_REPO_EDITED	AUDIT_SUCCESS	\${UserDisplayName} 已编辑存储库 \${ThinAppRepositoryName}，路径为 \${ThinAppRepositoryPath}
ADMIN_THINAPP_REPO_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除存储库 \${ThinAppRepositoryName}
ADMIN_UNREGISTER_PM	AUDIT_SUCCESS	\${UserDisplayName} 取消注册了物理机 \${MachineName}
ADMIN_UNREGISTER_PM_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能取消注册物理机 \${MachineName}
ADMIN_USER_INFO_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能为 \${UserName} 更新 AD 服务器中的用户信息
ADMIN_USER_INFO_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 为 \${UserName} 更新了 AD 服务器中的用户信息
ADMIN_USER_POLICY_DELETE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能为用户 \${UserName} 删除池 \${DesktopId} 的覆盖策略
ADMIN_USER_POLICY_DELETED	AUDIT_SUCCESS	\${UserDisplayName} 为用户 \${UserName} 删除了池 \${DesktopId} 的覆盖策略 (\${AttrChangeType}: \${AttrName} = \${AttrValue})
ADMIN_USER_POLICY_UPDATE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能为用户 \${UserName} 更新池 \${DesktopId} 的策略
ADMIN_USER_POLICY_UPDATED	AUDIT_SUCCESS	\${UserDisplayName} 为用户 \${UserName} 更新了池 \${DesktopId} 的策略 (\${AttrChangeType}: \${AttrName} = \${AttrValue})
ADMIN_USERLOGGEDIN	AUDIT_SUCCESS	用户 \${UserDisplayName} 已登录 View Administrator
ADMIN_USERLOGGEDOUT	AUDIT_SUCCESS	用户 \${UserDisplayName} 已从 View Administrator 注销
ADMIN_VC_ADD_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能添加 VC Server \${VCAddress}
ADMIN_VC_ADDED	AUDIT_SUCCESS	\${UserDisplayName} 已添加 VC Server \${VCAddress}



EventType	严重性	ModuleAndEventText
ADMIN_VC_EDITED	AUDIT_SUCCESS	\${UserDisplayName} 已编辑 VC Server \${VCAddress} (\${AttrChangeType}: \${AttrName} = \${AttrValue})
ADMIN_VC_LICINV_ALARM_DISABLED	AUDIT_SUCCESS	由于所有主机都具有桌面许可证，VC Server \${VCAddress} 上用于监视许可证清单的警报被禁用
ADMIN_VC_REMOVE_FAILED	AUDIT_FAIL	\${UserDisplayName} 未能移除 VC Server \${VCAddress}
ADMIN_VC_REMOVED	AUDIT_SUCCESS	\${UserDisplayName} 已移除 VC Server \${VCAddress}

## 事件消息属性

ModuleAndEventText 消息使用特定的属性。要确定某个属性的数据类型，您可以在 `event_data` 或 `event_data_historical` 表的 `type` 列中查看其值。

**表 2-7. ModuleAndEventText 消息使用的属性**

属性名称	说明
AdminFolderName	需要特别访问权的文件夹的名称。
AdminPermissionEntity	需要特别访问权的对象的名称。
AdminPrivilegeName	管理特权的名称。
AdminRoleName	管理角色的名称。
AdminRoleNewName	管理角色的新名称。
AttrChangeType	应用于通用属性的更改类型。
AttrName	通用属性的名称。
AttrValue	通用属性的值。
BrokerId	<b>View</b> 连接服务器实例的标识符。
BrokerName	<b>View</b> 连接服务器实例的名称。
DesktopDisplayName	桌面池的显示名称。
DesktopId	桌面池的标识符。
EntitlementDisplay	桌面授权的显示名称。
MachineId	物理机或虚拟机的名称。
MachineName	物理机或虚拟机的名称。
MaintenanceMode	维护模式状态。
ObjectID	清单对象的标识符。
ObjectType	清单对象的类型。
PolicyDisplayName	策略的显示名称。
PolicyObject	策略对象的标识符。

属性名称	说明
PolicyValue	策略对象的值。
ProtocolId	显示协议的标识符。
SecurityServerId	安全服务器的标识符。
SVIAdminFqdn	QuickPrep 域的 FQDN。
SVIAdminID	QuickPrep 域的标识符。
SVIAdminName	QuickPrep 域的名称。
SVIDeploymentGroupId	View Composer 部署组的标识符。
SVIOperation	View Composer 操作的名称。
SVIParentVM	View Composer 中的父虚拟机。
SVIPath	View Composer 中的对象路径。
SVISnapshot	View Composer 中的快照。
SVIVMID	View Composer 中虚拟机的标识符。
ThinAppDisplayName	ThinApp 对象的显示名称。
ThinAppId	ThinApp 对象的标识符。
ThinAppRepositoryName	ThinApp 存储库的名称
ThinAppRepositoryPath	ThinApp 存储库的路径。
时间	日期和时间值。
UserCount	24 个小时内桌面用户的最大数量。
UserDiskName	用户数据磁盘的名称。
UserDisplayName	DOMAIN\username 格式的用户名。
UserName	Active Directory 中的用户名。
VCAddress	vCenter Server 的 URL。

## 示例数据库查询和视图

您可以查询 `event_historical` 数据库，以显示错误事件、警告事件和特定的最新事件。

**注** 将以下示例中的 `dbo.VE_` 前缀替换为事件数据库的相应前缀。

### 列出错误事件

以下查询显示 `event_historical` 表中的所有错误事件。

```
CREATE VIEW error_events AS
(
    SELECT ev.EventID, ev.Time, ev.Module, ev.EventType, ev.ModuleAndEventText
    FROM dbo.VE_event_historical AS ev
    WHERE ev.Severity = 'ERROR'
);
```

## 列出警告事件

以下查询显示 `event_historical` 表中的所有警告事件。

```
CREATE VIEW warning_events AS
(
  SELECT ev.EventID, ev.Time, ev.Module, ev.EventType, ev.ModuleAndEventText
  FROM dbo.VE_event_historical AS ev
  WHERE ev.Severity = 'WARNING'
);
```

## 列出最新事件

以下查询列出与域 `MYDOM` 中用户 `fred` 关联的所有最新事件。

```
CREATE VIEW user_fred_events AS
(
  SELECT ev.EventID, ev.Time, ev.Module, ev.EventType, ev.Severity, ev.Acknowledged
  FROM dbo.VE_event_historical AS ev,
       dbo.VE_event_data_historical AS ed
  WHERE ev.EventID = ed.EventID AND ed.Name = 'UserDisplayName' AND ed.StrValue =
       'MYDOM\fred'
);
```

以下查询列出计算机上的代理关闭的所有最新事件。

```
CREATE VIEW agent_shutdown_events AS
(
  SELECT ev.EventID, ev.Time, ed.StrValue
  FROM dbo.VE_event_historical AS ev,
       dbo.VE_event_data_historical AS ed
  WHERE ev.EventID = ed.EventID AND ev.EventType = 'AGENT_SHUTDOWN' AND
       ed.Name = 'MachineName'
);
```

以下查询列出因桌面池为空而导致桌面启动失败的所有最新事件。

```
CREATE VIEW desktop_launch_failure_events AS
(
  SELECT ev.EventID, ev.Time, ed1.StrValue, ed2.StrValue
  FROM dbo.VE_event_historical AS ev,
       dbo.VE_event_data_historical AS ed1,
       dbo.VE_event_data_historical AS ed2
  WHERE ev.EventID = ed1.EventID AND ev.EventID = ed2.EventID AND
       ev.EventType = 'BROKER_POOL_EMPTY' AND
       ed1.Name = 'UserDisplayName' AND ed2.Name = 'DesktopId'
);
```

以下查询列出管理员移除了桌面池的所有最新事件。

```
CREATE VIEW desktop_pool_removed_events AS
(
  SELECT ev.EventID, ev.Time, ed1.StrValue, ed2.StrValue
```

```

FROM dbo.VE_event_historical AS ev,
     dbo.VE_event_data_historical AS ed1,
     dbo.VE_event_data_historical AS ed2
WHERE ev.EventID = ed1.EventID AND ev.EventID = ed2.EventID AND
      ev.EventType = 'ADMIN_DESKTOP_REMOVED' AND
      ed1.Name = 'UserDisplayName' AND ed2.Name = 'DesktopId'
);

```

以下查询列出管理员添加了 **ThinApp** 存储库的所有最新事件。

```

CREATE VIEW thinapp_repository_added_events AS
(
  SELECT ev.EventID, ev.Time, ed1.StrValue, ed2.StrValue, ed3.StrValue
  FROM dbo.VE_event_historical AS ev,
       dbo.VE_event_data_historical AS ed1,
       dbo.VE_event_data_historical AS ed2,
       dbo.VE_event_data_historical AS ed3
  WHERE ev.EventID = ed1.EventID AND ev.EventID = ed2.EventID AND ev.EventID = ed3.EventID
  AND
       ev.EventType = 'ADMIN_THINAPP_REPO_ADDED' AND
       ed1.Name = 'UserDisplayName' AND ed2.Name = 'ThinAppRepositoryName' AND
       ed3.Name = 'ThinAppRepositoryPath'
);

```

# 使用 View PowerCLI

自 Horizon 7 版本 7.0.3 开始，View PowerCLI 已弃用。

对于以前版本的 Horizon 7，您可以使用 View PowerCLI 作为 View 的一个易于使用的 PowerShell 界面，并使用 View PowerCLI cmdlet 对 View 组件执行各种管理任务。

在 Horizon 7 版本 7.0.3 中，可以通过 VMware PowerCLI 使用 Horizon PowerCLI cmdlet。使用 Horizon PowerCLI cmdlet 对 Horizon 组件执行各种管理任务。

有关 Horizon PowerCLI cmdlet 的更多信息，请参阅《VMware PowerCLI cmdlet 参考》。

有关创建高级函数和脚本以用于 Horizon PowerCLI 的 API 规范的信息，请参阅 [VMware 开发人员中心](#) 的 View API 参考。

有关可用于创建您自己的 Horizon PowerCLI 脚本的示例脚本的更多信息，请访问 [GitHub 上的 Horizon PowerCLI 社区](#)。

本章讨论了以下主题：

- [View PowerCLI 使用入门](#)
- [View Administrator、PowerCLI Cmdlet 和 View 命令行界面比较](#)
- [View PowerCLI cmdlet 参考](#)
- [View PowerCLI cmdlet 参数](#)
- [使用 View PowerCLI cmdlet 的示例](#)
- [使用 View PowerCLI 执行高级任务的示例](#)
- [向桌面池分配多个网络标签](#)

## View PowerCLI 使用入门

PowerShell 是专为 Microsoft Windows 设计的命令行和脚本环境。PowerShell 使用 .NET 对象模型，为管理员提供管理和自动化功能。您可以通过运行在 PowerShell 中被称为 cmdlet 的命令来使用 PowerShell。View PowerCLI cmdlet 的命令行语法与常规的 PowerShell 语法相同。

View PowerCLI cmdlet 在 PowershellServiceCmdlets.dll 文件中进行定义，该文件安装在 C:\Program Files\VMware\VMware View\Server\bin 目录中。PowershellServiceCmdlets.dll 文件构成 VMware.View.Broker 管理单元。

您可以编辑和扩展 View PowerCLI cmdlet 的脚本配置文件 InitViewCmdlets.ps1，以定义 cmdlet 别名，配置环境并设置启动操作。InitViewCmdlets.ps1 位于 View 安装目录下的 Extras 文件夹中。

您可以将 View PowerCLI cmdlet 与 vSphere PowerCLI cmdlet 结合使用。vSphere PowerCLI cmdlet 为 VMware vSphere 提供了一个管理接口。如果 vSphere PowerCLI 安装在 View 连接服务器实例上，则 vSphere PowerCLI cmdlet 在启动 View PowerCLI 时加载。

在 View PowerCLI 中，您可以通过 ID 引用虚拟机或 vCenter Server 实例，但无法将这些实体作为对象进行传递。对于其他 vSphere 对象（如资源池和文件夹），您必须提供完整路径。您可以使用 View PowerCLI cmdlet 检查 View 中的 vCenter Server 实例配置。

有关使用 PowerShell 的常规信息，请参阅 Microsoft 文档。

## 启动加载了 View PowerCLI 的 PowerShell 控制台

您可以在 View 连接服务器主机上直接运行 View PowerCLI cmdlet。

### 前提条件

确认 View 连接服务器主机已安装了以下软件。

- View 4.5 或更高版本
- Microsoft .NET Framework
- Windows PowerShell 1.0

### 步骤

- 1 以拥有足够特权修改配置数据的用户角色登录到 View 连接服务器实例。  
例如，管理员角色可以修改配置数据。只读角色无法更新配置数据。
- 2 选择开始 > 所有程序 > VMware > View PowerCLI。

### 后续步骤

如果收到错误消息，指出因为禁止执行脚本而无法加载脚本配置文件，请键入 PowerShell Set-ExecutionPolicy Unrestricted 命令，然后重新启动 PowerShell 控制台。

## 从自远程系统使用 View PowerCLI cmdlet

您可以使用 PowerShell 远程功能从远程系统访问 View PowerCLI cmdlet。

### 步骤

- 1 在远程系统上，使用文本编辑器打开 C:\Windows\System32\WindowsPowerShells\v1.0\Profile.ps1 文件。
- 2 将行 `add-pssnapin vm*` 添加到 Profile.ps1 文件中。
- 3 保存更改。

将在远程系统上向 PowerShell 配置文件添加 View PowerCLI 管理单元。

### 后续步骤

对其他远程 PowerShell 操作采取保护 View PowerCLI 操作的相同预防措施。

## 显示 View PowerCLI 的帮助

您可以通过在 PowerShell 控制台中键入命令来显示 View PowerCLI 的帮助。

### 步骤

- 1 在 View 连接服务器实例上，选择开始 > 所有程序 > VMware > View PowerCLI。
- 2 显示 View PowerCLI 的帮助。

选项	操作
列出所有 View PowerCLI cmdlet	运行 Get-Command cmdlet。 例如: <code>Get-Command -PSSnapin VMware.View.Broker   more</code>
显示特定 cmdlet 的帮助	键入 Get-Help, 后跟 cmdlet 的名称。 例如: <code>Get-Help Add-ViewVC   more</code>
显示特定 cmdlet 的详细帮助	键入 Get-Help, 后跟 cmdlet 的名称和 -full 参数。 例如: <code>Get-Help Add-ViewVC -full   more</code> 或者, 使用 Get-Help 的 help 别名。 例如: <code>Add-ViewVC -full   more</code>

## 检查 View PowerCLI cmdlet 错误

View PowerCLI cmdlet 将所有错误作为非终止性错误来处理, 这类错误会导致 cmdlet 暂停执行, 但不会终止管线。您可以检查 \$error 自动变量来确定错误的原因。

要控制 PowerShell 处理非终止性错误的方式以及在 shell 中显示错误的方式, 请设置标准 PowerShell \$ErrorActionPreference 和 \$ErrorView 自动变量。

## 通过管道传输并指定相同类型的对象

如果您尝试将对象通过管道传输到 cmdlet 并为该 cmdlet 指定一个相同类型的对象, 则此 cmdlet 将失败并显示以下错误。

```
The input object cannot be bound to any parameters for the command either because the command does not take pipeline input or the input and its properties do not match any of the parameters that take pipeline input.
```

例如, 以下 cmdlet 用法会生成此错误。

```
Get-Pool -pool_id Pool1 | Update-ManualPool -pool_id Pool2 -displayName "Manual Pool 2"
```

## 转义 vCenter Server 路径名称中的字符

如果您指定的 vCenter Server 文件夹路径中有实体名称包含特殊字符，您必须对特殊字符进行转义。

**表 3-1. 特殊字符的转义序列**

特殊字符	转义序列
%	%25
/	%2f
\	%5c

请勿对路径名称本身含有的斜线进行转义。例如，将文件夹路径 `/datacenter_01/vm/img%-12` 表示为 `/datacenter_01/vm/img%25-12`。

某些 cmdlet 和参数需要在实体名称中使用转义序列。

**表 3-2. 需要使用转义序列的 cmdlet 参数**

cmdlet	需要使用转义序列的参数
Add-AutomaticLinkedClonePool	-datastoreSpecs
Update-AutomaticLinkedClonePool	-parentVMPath -resourcePoolPath -vmfolderPath
Add-AutomaticPool	-datastorePaths
Update-AutomaticPool	-resourcePoolPath -templatePath -vmfolderPath
Send-LinkedCloneRecompose	-parentVMPath

## View Administrator、PowerCLI Cmdlet 和 View 命令行界面比较

您可以使用 View Administrator、View PowerCLI cmdlet、vdmadmin、vdmutil 和 lmvutil 对 View 对象执行管理操作。并非所有管理操作在所有界面中都可执行。

**表 3-3. View Administrator、PowerCLI 和 View 命令行界面操作**

对象	操作	View Administrator	View PowerCLI	vdmadmin	vdmutil 或 lmvutil
应用程序池	添加	X			
	删除	X			
	更新	X			
	授权用户	X			
	移除授权	X			
Cloud Pod 架构	查看 Cloud Pod 架构配置	X			X



对象	操作	View Administrator	View PowerCLI	vdmadmin	vdmutil 或 lmvutil
	查看容器联合运行状况	X			
	列出全局授权	X			X
	列出全局授权中的池（桌面或应用程序）	X			X
	列出全局授权中的用户和组	X			X
	列出用户的有效主站点	X			X
	列出专用桌面池分配	X			X
	列出 Cloud Pod 架构拓扑中的容器或站点	X			X
	在容器联合中查看桌面和应用程序会话	X			X
桌面池	添加	X	X		
	分配专用桌面池			X	
	分配 ThinApp	X			
	禁用	X	X		
	启用	X	X		
	授权用户	X	X		
	获取信息	X	X		
	获取未授权的策略	X		X	
	获取未授权的用户	X		X	
	移除	X	X		
	移除分配			X	
	移除授权	X	X		
	限制授权	X			
	设置策略	X	X		
	更新	X	X		
	分配网络标签		X		
	获取网络标签配置		X		
域筛选器	获取信息			X	
	移除筛选器			X	
	设置筛选器			X	
事件	获取列表	X	X	X	
	获取报告		X	X	
场	添加	X			
	禁用	X			

对象	操作	View Administrator	View PowerCLI	vdmadmin	vdmutil 或 lmvutil
文件夹	启用	X			
	获取信息	X			
	删除	X			
	更新	X			
	添加	X			
	获取信息	X			
	移动	X			
	移除	X			
全局授权（桌面或应用程序）	添加	X			X
	更新	X			X
	移除	X			X
	添加池	X			X
	移除池	X			X
	添加用户或组	X			X
	移除用户或组	X			X
主站点	将主站点分配给用户或组	X			X
	移除	X			X
	修改主站点分配	X			X
	创建主站点覆盖	X			X
	修改主站点覆盖	X			X
	移除主站点覆盖	X			X
	列出用户的有效主站点	X			X
即时克隆桌面	推送映像	X			
	恢复	X			
即时克隆域管理员	添加	X			
	获取信息	X			
	移除	X			
	更新	X			
Kiosk 模式	添加客户端帐户			X	
	禁用身份验证			X	
	启用身份验证			X	
	获取信息			X	
	获取默认值			X	
	移除客户端帐户			X	

对象	操作	View Administrator	View PowerCLI	vdmadmin	vdmutil 或 lmvutil
	设置默认值			X	
链接克隆桌面	重新平衡	X	X		
	重构	X	X		
	重新创建	X			
	刷新	X	X		
	还原	X			
	设置存储过量置备	X	X		
	获取网络标签配置		X		
链接克隆场	重构	X			
	设置存储过量置备	X			
权限	添加	X			
	获取信息	X			
	移除	X			
永久用户数据磁盘	附加	X			
	删除	X			
	分离	X			
	获取信息	X	X		
	Replace（替换）	X			
安装了 Horizon Agent 的物理机	获取信息	X	X	X	
容器	初始化	X			X
	取消初始化	X			X
	加入	X			X
	取消加入	X			X
	更新	X			X
远程会话	断开连接	X	X		
	获取信息	X	X		
	注销	X	X		
角色	添加	X			
	修改	X			
	移除	X			
RDS 主机	获取信息	X	X		
SAML 身份验证器	添加	X			
	获取信息	X			
	移除	X			

对象	操作	View Administrator	View PowerCLI	vdmadmin	vdmutil 或 lmvutil
站点	更新	X			
	添加	X			X
	更新	X			X
	移除	X			X
	向站点分配容器	X			X
	从站点中移除容器	X			X
用户	配置策略	X			
	创建管理员	X			
	获取信息	X	X	X	
	移除管理员	X			
	更新 FSP			X	
vCenter Server 实例	添加	X	X		
	获取信息	X	X		
	移除	X	X		
	更新	X	X		
Horizon Agent	创建 DCT 捆绑包			X	
	获取日志文件的副本			X	
	获取日志文件的列表			X	
	获取日志级别			X	
	获取状态			X	
	获取版本			X	
	覆盖 IP 地址			X	
	设置日志级别			X	
View Composer 域	获取信息		X		
View 连接服务器实例	备份配置	X			
	获取信息	X	X		
	从组中移除			X	
	还原配置	X			
	更新	X	X		
View 连接服务器组	设置组的 GUID			X	
	设置组的名称			X	
View 全局设置	获取信息	X	X		
	更新	X	X		
View 服务运行状况监视器	获取信息	X	X	X	

对象	操作	View Administrator	View PowerCLI	vdmadmin	vdmutil 或 Imvutil
安装了 Horizon Agent 的虚拟机	获取信息	X	X	X	
	移除所有权	X	X		
	重置	X	X		
	更新所有权	X	X		
VMware Horizon 许可证	获取信息	X	X		
	设置许可证	X	X		

## View PowerCLI cmdlet 参考

您可以使用 View PowerCLI cmdlet 来管理 View 连接服务器实例中的 View。

下表列出了所有可用的 View PowerCLI cmdlet，这些 cmdlet 按 View 对象进行组织。有关 cmdlet 语法，请使用 Get-Help cmdlet。有关更多信息，请参阅[显示 View PowerCLI 的帮助](#)。

**表 3-4. View PowerCLI cmdlet**

对象	cmdlet	说明
桌面池	Get-Pool	返回有关桌面池的信息。
	Remove-Pool	移除桌面池。
	Add-PoolEntitlement	为用户创建桌面池授权。
	Get-PoolEntitlement	返回有关有权使用桌面池的用户的信息。
	Remove-PoolEntitlement	移除用户的桌面池授权。
链接克隆桌面池	Add-AutomaticLinkedClonePool	添加自动置备的链接克隆桌面池。
	Update-AutomaticLinkedClonePool	更新自动置备的链接克隆桌面池。
	Send-LinkedCloneRebalance	在可用的逻辑驱动器之间重新平衡链接克隆桌面。
	Send-LinkedCloneRecompose	从父虚拟机快照重构链接克隆桌面。
	Send-LinkedCloneRefresh	将链接克隆桌面的操作系统磁盘刷新到它们的原始状态和大小。
完整克隆桌面池	Export-NetworkLabelSpecForLinkedClone	列出要部署链接克隆桌面池的指定群集中所有主机上的共享网络标签。输出将导出到配置文件。
	Export-NetworkLabelSpecForFullClone	列出要部署完整克隆桌面池的指定群集中所有主机上的共享网络标签。输出将导出到配置文件。
自动虚拟机桌面池	Add-AutomaticPool	添加自动置备的完整虚拟机桌面池。
	Update-AutomaticPool	更新自动置备的完整虚拟机桌面池。
手动桌面池	Add-ManualPool	添加手动置备的受管桌面池。
	Update-ManualPool	更新手动置备的受管桌面池。
	Add-ManualUnmanagedPool	添加手动置备的未受管桌面池。

对象	cmdlet	说明
	Update-ManualUnmanagedPool	更新手动置备的未受管桌面池。
View Composer 域	Get-ComposerDomain	返回有关 View Composer 的信息。
View 连接服务器实例	Get-ConnectionBroker	返回有关 View 连接服务器和安全服务器实例的信息。
	Update-ConnectionBroker	更新 View 连接服务器或安全服务器实例的配置。
VMware Horizon 许可证	Get-License	返回 View 连接服务器实例中的 View 许可证。
	Set-License	设置 View 连接服务器实例中的 View 许可证。
物理机	Get-DesktopPhysicalMachine	返回可在未受管桌面池中使用的物理机列表。
虚拟机	Get-DesktopVM	返回有关虚拟机的信息。
	Send-VMReset	重置虚拟机。
事件	Get-EventReport	返回指定视图的事件报告。
	Get-EventReportList	返回可用于 Get-EventReport cmdlet 的视图。
View 全局设置	Get-GlobalSetting	返回有关 View 环境的全局配置信息。
	Update-GlobalSetting	更新有关 View 环境的全局配置信息。
View 服务运行状况监视器	Get-Monitor	返回 View 服务的运行状况监视器列表。
永久用户数据磁盘	Get-ProfileDisk	返回有关永久用户数据磁盘的信息。
远程会话	Get-RemoteSession	返回有关活动的远程会话的信息。
	Send-SessionDisconnect	断开活动的远程会话连接。
	Send-SessionLogoff	注销活动的远程会话。
用户	Get-User	返回有关用户的信息。
	Remove-UserOwnership	移除虚拟机的所有权。
	Update-UserOwnership	将用户（指定为 SID）分配给虚拟机。此 cmdlet 不支持将用户分配给物理机。
vCenter Server 实例	Add-ViewVC	将 vCenter Server 实例添加到 View。
	Get-ViewVC	返回有关 vCenter Server 实例的信息。
	Remove-ViewVC	从 View 中移除 vCenter Server 实例。
	Update-ViewVC	更新 View 中的 vCenter Server 实例的配置。

## View PowerCLI cmdlet 参数

某些 View PowerCLI cmdlet 参数接受设置。例如，`-flashQuality` 参数接受为 Adobe Flash 内容指定允许的最高质量的设置。

### 默认显示协议参数

`-defaultProtocol` 参数用于指定桌面池的默认显示协议。

表 3-5. -defaultProtocol 参数设置

设置	说明
PCoIP	将默认显示协议设置为 PCoIP。
RDP	将默认显示协议设置为 Microsoft RDP。

## 删除策略参数

-deletePolicy 参数用于指定自动置备的浮动和链接克隆桌面池的删除策略。

表 3-6. -deletePolicy 参数设置

设置	说明
默认	用户注销时不删除计算机。
DeleteOnUse	用户注销时删除计算机。
RefreshOnUse	用户注销时刷新计算机。
<b>注</b> 此设置仅适用于链接克隆桌面池。	

## Flash 质量参数

-flashQuality 参数用于为 Adobe Flash 内容指定允许的最高质量。此值会覆盖网页上的设置。如果某网页的 Adobe Flash 质量高于允许的最大值，客户端会将其降低至指定的最高质量。降低 Adobe Flash 内容的质量可减少内容使用的带宽。

表 3-7. -flashQuality 参数设置

设置	说明
高	允许低、中或高质量 Flash 内容。
低	仅允许低质量 Flash 内容。
中	允许低质量或中质量 Flash 内容。
NO_CONTROL	让网页设置来确定 Flash 内容的质量。

## Flash 限制参数

-flashThrottling 参数用于指定 Adobe Flash 刷新屏幕上信息的频率。限制 Adobe Flash 增加刷新时间间隔会降低帧速率。这种降低帧速率的方法可减少 Adobe Flash 内容使用的带宽，但也可能会导致 Adobe Flash 丢帧。

表 3-8. -flashThrottling 参数设置

设置	说明
AGGRESSIVE	将刷新时间间隔设置为 2500 毫秒。此设置产生最高数量的丢帧。音频传输速度不受影响。
CONSERVATIVE	将刷新时间间隔设置为 100 毫秒。此设置产生的丢帧数最低。音频传输速度不受影响。

设置	说明
DISABLED	禁用限制功能。不修改计时器间隔。
MODERATE	将刷新间隔设置为 500 毫秒。音频传输速度不受影响。

## LDAP 备份频率参数

`-ldapBackupFrequency` 参数用于指定 View 连接服务器实例的 LDAP 备份频率。

**表 3-9. -ldapBackupFrequency 参数设置**

设置	说明
Every12Hour	每 12 小时备份一次 LDAP 数据库。
Every2Day	每 2 天备份一次 LDAP 数据库。
Every2Week	每 2 周备份一次 LDAP 数据库。
Every6Hour	每 6 小时备份一次 LDAP 数据库。
EveryDay	每天备份一次 LDAP 数据库。
EveryHour	每小时备份一次 LDAP 数据库。
EveryWeek	每周备份一次 LDAP 数据库。
从不	对 LDAP 数据库关闭备份功能。

## 池类型参数

`-poolType` 参数用于指定桌面池类型。

**表 3-10. -poolType 参数设置**

设置	说明
IndividualUnmanaged	池中包含单个未受管的计算机。
IndividualVC	池中包含由 vCenter Server 实例管理和配置的单个计算机。
手动	池中包含由 vCenter Server 实例管理和配置的手动配置的浮动（非持久）计算机。
ManualUnmanagedNonPersistent	池中包含未由 vCenter Server 实例管理的手动配置的浮动（非持久）计算机。
ManualUnmanagedPersistent	池中包含未由 vCenter Server 实例管理的手动配置的专用（持久）计算机。
ManualVCPersistent	池中包含由 vCenter Server 实例管理的手动配置的专用（持久）计算机。
NonPersistent	(AutomaticPool) 池中包含由 vCenter Server 实例置备、管理和配置的自动配置的浮动（非持久）计算机。
OnRequestSviNonPersistent	(AutomaticPool) 池中包含请求时由 vCenter Server 实例和 View Composer 置备、管理和配置的浮动（非持久）计算机。
OnRequestSviPersistent	(AutomaticPool) 池中包含请求时由 vCenter Server 实例和 View Composer 置备、管理和配置的专用（持久）计算机。
OnRequestVcNonPersistent	(AutomaticPool) 池中包含请求时由 vCenter Server 实例置备、管理和配置的浮动（非持久）计算机。



设置	说明
OnRequestVcPersistent	(AutomaticPool) 池中包含请求时由 vCenter Server 实例置备、管理和配置的专用（持久）计算机。
Persistent	(AutomaticPool) 池中包含由 vCenter Server 实例置备、管理和配置的自动配置的专用（持久）计算机。
SVINonPersistent	(AutomaticPool) 池中包含由 vCenter Server 实例和 View Composer 置备、管理和配置的浮动（非持久）计算机。
SVIPersistent	(AutomaticPool) 池中包含由 vCenter Server 实例和 View Composer 置备、管理和配置的专用（持久）计算机。

## 电源策略参数

`-powerPolicy` 参数用于指定桌面池的电源策略。

**表 3-11. -powerPolicy 设置**

设置	说明
AlwaysOn	配置计算机使其保持电源打开状态，即使在没有用户使用也是如此。如果关闭计算机，该计算机将会立即重新启动。
RemainOn	在需要时启动已关闭的计算机。计算机保持电源打开状态，直到将其关闭时为止。
PowerOff	关闭无人使用的计算机。
挂起	挂起无人使用的计算机。

## 刷新策略类型参数

`-refreshPolicyType` 参数用于指定自动置备的专用和链接克隆桌面池的操作系统磁盘的刷新策略。

**表 3-12. -refreshPolicyType 设置**

设置	说明
始终	每当用户注销时刷新操作系统磁盘。
Conditional	仅当满足特定条件时，才会在用户注销后刷新操作系统磁盘。使用 <code>-refreshpolicyDays</code> 和 <code>-refreshPolicyUsage</code> 参数指定刷新时间间隔（以天为单位）和允许的最大磁盘大小百分比。
从不	在用户注销时从不刷新操作系统磁盘。

## 智能卡设置参数

`-smartCardSetting` 参数用于指定智能卡身份验证策略设置。此设置适用于 `Update-ConnectionBroker`，并且除非还使用 `Update-GlobalSetting` 指定了 `-UseSSLClient $true`，否则此设置不会生效。

表 3-13. -smartCardSetting 参数设置

设置	说明
NotAllowed	禁用智能卡身份验证。
可选	允许用户使用智能卡身份验证或密码身份验证连接到该 View 连接服务器实例。如果智能卡身份验证失败，用户就必须提供密码。
需要	要求用户在连接到 View 连接服务器实例时使用智能卡身份验证。智能卡身份验证仅替代 Windows 密码身份验证。如果已启用 SecureID，用户就必须同时使用 SecureID 和智能卡身份验证机制进行身份验证。

## View Composer 任务参数

-composerTask 参数用于在虚拟机上指定 View Composer 维护任务。

表 3-14. -composerTask 参数设置

设置	说明
attachUdd	附加永久磁盘。
detachUdd	分离永久磁盘。
mkChkPoint	创建检查点快照。
重新平衡	重新平衡链接克隆计算机。
replaceUdd	替换永久磁盘。
重新同步	重构链接克隆计算机。

## 使用 View PowerCLI cmdlet 的示例

使用 View PowerCLI cmdlet，您可以从命令行或脚本执行 View 管理任务，而不使用 View Administrator。

### 管理 View 连接服务器实例

您可以使用 View PowerCLI cmdlet 执行 View 连接服务器管理任务。

表 3-15. 常见 View 连接服务器管理任务示例

任务	View PowerCLI cmdlet 语法示例
获取特定 View 连接服务器实例的配置设置	Get-ConnectionBroker -broker_id CONNSVR1
更新特定 View 连接服务器实例的配置设置	Update-ConnectionBroker -broker_id CONNSVR1 -directConnect \$false -secureIdEnabled \$true -ldapBackupFrequency EveryWeek
配置特定 View 连接服务器实例的安全 PCoIP 连接	Update-ConnectionBroker -broker_id CS-VSG -directPCoIP \$FALSE
设置特定 View 连接服务器实例的 PCoIP 外部 URL	Update-ConnectionBroker -broker_id CS-VSG -externalPCoIPURL 10.18.133.34:4172
设置特定安全服务器的 PCoIP 外部 URL	Update-ConnectionBroker -broker_id SECSVR-03 -externalPCoIPURL 10.116.32.136:4172

## 在 View 中管理 vCenter Server 实例

您可以使用 View PowerCLI cmdlet 在 View 中执行 vCenter Server 管理任务。

**表 3-16. View 中常见 vCenter Server 管理任务示例**

任务	View PowerCLI cmdlet 语法示例
将 vCenter Server 实例添加到 View 配置	<code>Add-ViewVC -serverName vc01.mydom.int -username Administrator -password clydenw -createRampFactor 5 -deleteRampFactor 5</code>
获取 View 中特定 vCenter Server 实例的相关信息	<code>Get-ViewVC -serverName vc01.mydom.int</code>
获取特定 DNS 域中所有 vCenter Server 实例的相关信息	<code>Get-ViewVC -serverName *.mycorp.com</code>
更改特定 vCenter Server 实例的负载增加因素值	<code>Get-ViewVC -serverName svr11.mycorp.com   Update-ViewVC -createRampFactor 5 -deleteRampFactor 10</code>
更改特定 DNS 域中所有 vCenter Server 实例的创建负载增加因素值	<code>Get-ViewVC -serverName *.mycorp.com   Update-ViewVC -createRampFactor 5</code>
从 View 配置中移除 vCenter Server 实例	<code>Get-ViewVC -serverName vc02.mydom.int   Remove-ViewVC</code>

## 管理桌面池

您可以使用 View PowerCLI cmdlet 执行桌面池管理任务。

**表 3-17. 常见桌面池管理任务示例**

任务	View PowerCLI cmdlet 语法示例
获取具有特定显示名称的桌面池的相关信息	<code>Get-Pool -displayName "My Pool 1"</code>
获取显示名称中包含特定前缀的所有桌面池的相关信息	<code>Get-Pool -pool_id mypool-*</code>
获取已配置为使用 PCoIP 显示协议的所有桌面池的相关信息	<code>Get-Pool -protocol PCoIP</code>
获取所有单个未受管桌面池的相关信息	<code>Get-Pool -poolType IndividualUnmanaged</code>
移除具有特定池 ID 的桌面池	<code>Remove-Pool -pool_id dtpool-10</code>
移除具有特定池 ID 的桌面池并终止任何活动的会话，但不从磁盘中删除其映像	<code>Remove-Pool -pool_id dtpool-12 -TerminateSession \$true -DeleteFromDisk \$false</code>
移除具有特定显示名称的桌面池并从磁盘中删除其映像	<code>Get-Pool -displayName "My Pool 1"   Remove-Pool -DeleteFromDisk \$true</code>

## 创建和更新自动置备的桌面池

您可以使用 `Get-ViewVC` 和 `Update-AutomaticPool` cmdlet 创建和更新自动置备的桌面池。

在以下示例中，`Get-ViewVC` cmdlet 添加名为 `auto1` 的自动置备桌面池。该桌面池由名为 `vc.mydom.int` 的 vCenter Server 实例管理。

```
Get-ViewVC -serverName vc.mydom.int | Add-AutomaticPool -pool_id auto1 -displayName "ADP1"
-namePrefix "adp1-{n:fixed=4}" -vmFolderPath /AutoConfig/vm
-resourcePoolPath /AutoConfig/host/Resources -templatePath /AutoConfig/vm/ADP_template
-dataStorePaths /host/datastore_1/lun10 -customizationSpecName "Windows 7 Variation 3"
-minimumCount 4 -maximumCount 10
```

您可以通过将 `-minimumCount` 和 `-maximumCount` 参数设置为相同的值，预先置备所有桌面。

在以下示例中，`Update-AutomaticPool` cmdlet 更新自动置备桌面池 `auto1` 的配置。

```
Update-AutomaticPool -pool_id auto1 -displayName "Automatic Desktop Pool 1"
-isProvisioningEnabled $false -dataStorePaths /host/datastore_1/lun10;/host/datastore_1/lun12
```

由于在 `-dataStorePaths` 参数中指定的数据存储覆盖了之前的设置，您必须在参数中指定任何现有数据存储，以供桌面池继续使用这些数据存储。

## 创建和更新链接克隆桌面池

您可以使用 `Get-ViewVC` 和 `Get-DesktopVM` cmdlet 创建和更新链接克隆桌面池。

在以下示例中，`Get-ViewVC` cmdlet 添加名为 `lcdpool_1` 的链接克隆桌面池。该池由名为 `vc.mydom.int` 的 vCenter Server 实例上的 View Composer 管理。

```
Get-ViewVC -serverName vc.mydom.int | Get-ComposerDomain -domain VCDOM |
Add-AutomaticLinkedClonePool -pool_id lcdpool_1 -displayName "LCD Pool 1"
-namePrefix "lcp1-{n}-dt" -parentVMPATH /AutoPoolVMs/parent
-parentSnapshotPath /AutoPoolSnapshots/parent1_snapshot -vmFolderPath /AutoConfig/VM_folder
-resourcePoolPath /AutoConfig/host/Resources
-datastoreSpecs [Aggressive,os,data]/host/datastore_1/lun04;/host/datastore_2/lun16
-dataDiskLetter "D" -dataDiskSize 100 -minimumCount 4 -maximumCount 10
```

您可以通过将 `-minimumCount` 和 `-maximumCount` 参数设置为相同的值，预先置备所有桌面。如果您指定永久数据磁盘，请为驱动器使用大写字母。请勿使用父虚拟机上已经存在的驱动器盘符（如 `A`、`B` 或 `C`）或与网络挂载驱动器相冲突的盘符。

在以下示例中，`Get-ViewVC` cmdlet 更新链接克隆桌面池 `lcdpool_1` 的配置。

```
Get-ViewVC -serverName vc.mydom.int | Get-ComposerDomain -domain VCDOM |
Update-AutomaticLinkedClonePool -pool_id lcdpool_1 -datastoreSpecs
[Conservative,os,data]/host/datastore_1/lun04;/host/datastore_2/lun16;/host/datastore_2/lun22
-minimumCount 4 -maximumCount 20 -headroomCount 2 -powerPolicy Suspend -defaultProtocol PCoIP
-isUserResetAllowed $true
```

由于在 `-datastoreSpecs` 参数中指定的数据存储覆盖了之前的设置，您必须在参数中指定任何现有数据存储，以供该池继续使用这些数据存储。

您可以使用 `Get-DesktopVM cmdlet` 执行重新平衡、刷新和重构操作。

**表 3-18. 重新平衡、刷新和重构操作示例**

任务	View PowerCLI cmdlet 语法示例
在链接克隆桌面池中的可用数据存储之间重新平衡桌面	<code>Get-DesktopVM -pool_id lcdpool_2   Send-LinkedCloneRebalance -schedule 2011-05-10:01:00:00 -forceLogoff \$false -stopOnError \$true</code>
通过将每个链接克隆桌面中的操作系统磁盘恢复为原始状态和大小，对其进行刷新	<code>Get-DesktopVM -pool_id lcdpool_2   Send-LinkedCloneRefresh -schedule "May 12 2011 01:15" -forceLogoff \$true -stopOnError \$true</code>
通过父虚拟机快照重构所有链接克隆桌面	<code>Get-DesktopVM -pool_id lcdpool_2   Send-LinkedCloneRecompose -schedule ((Get-Date).AddHours(8)) -parentVMPath /AutoPoolVMs/parent2 -parentSnapshotPath /AutoPoolSnapshots/parent2_snapshot -forceLogoff \$true -stopOnError \$true</code>

## 创建和更新手动置备的桌面池

您可以使用 `Add-ManualPool`、`Get-ViewVC` 和 `Update-ManualPool cmdlet` 创建和更新手动置备的桌面池。

在以下示例中，`Add-ManualPool cmdlet` 创建名为 `manPool` 的手动置备桌面池，其中包含名为 `myVM` 的虚拟机。

```
Add-ManualPool -pool_id manPool -id (Get-VM -name "myVM").id -isUserResetAllowed $true
```

在以下示例中，`Get-ViewVC cmdlet` 通过由名为 `vc.mydom.int` 的 vCenter Server 实例管理的桌面，创建名为 `man1` 的手动置备桌面池。

```
Get-ViewVC -serverName vc.mydom.int | Get-DesktopVM -poolType Manual | Add-ManualPool -pool_id man1 -isUserResetAllowed $false
```

在以下示例中，`Update-ManualPool cmdlet` 更新手动置备桌面池 `man1` 的配置。

```
Update-ManualPool -pool_id man1 -displayName "Manual Desktop 1" -isUserResetAllowed $true
```

**注** 要使用 `Get-VM cmdlet`，您必须安装 vSphere PowerCLI。

## 创建和更新手动未受管桌面池

您可以使用 `Add-ManualUnmanagedPool` 和 `Update-ManualUnmanagedPool cmdlet` 创建和更新手动未受管桌面池。

在以下示例中，`Add-ManualUnmanagedPool cmdlet` 创建名为 `unman1` 的未受管桌面池，其中包含名为 `pm01` 和 `pm02` 的物理机。

```
Add-ManualUnmanagedPool -pool_id unman1 -pm_id_list pm01;pm02 -isUserResetAllowed $true
```

在以下示例中，Update-ManualUnmanagedPool cmdlet 更新未受管桌面池 unman1 的配置。

```
Update-ManualUnmanagedPool -pool_id unman1 -displayName "Unmanaged Desktop 1"
-isUserResetAllowed $false
```

## 显示有关用户和组的信息

您可以使用 Get-User cmdlet 显示有关 Active Directory 用户和组的信息。

在以下示例中，Get-User cmdlet 显示位于 mydom 域中的所有用户的相关信息。

```
Get-User -domain "mydom"
```

在以下示例中，Get-User cmdlet 显示位于 mydom 域中的名为 fred 的用户的相关信息。不包括有关用户组的信息。

```
Get-User -name "fred" -domain "mydom" -includeGroup $false
```

## 管理桌面授权

您可以使用 View PowerCLI cmdlet 来管理桌面授权。

**表 3-19. 常见桌面授权管理任务示例**

任务	View PowerCLI cmdlet 语法示例
授权特定域中的用户使用特定桌面池	Get-User -name "mydom\fred"   Add-PoolEntitlement -pool_id dtop-12
授权用户使用所有桌面池	Get-Pool   Add-PoolEntitlement -sid (Get-User -name "usr1").sid
获取有关有权使用特定桌面池的所有用户的信息	Get-PoolEntitlement -pool_id dtop-1
获取有关有权使用 ID 具有特定前缀的桌面池的所有用户的信息	Get-Pool -pool_id dtpool-*   Get-PoolEntitlement
移除使用特定桌面池的授权	Get-PoolEntitlement -pool_id dtpool-11   Remove-PoolEntitlement
移除所有授权	Get-PoolEntitlement   Remove-PoolEntitlement -forceRemove \$true

**注** 如果未指定 -forceRemove 参数，您可以使用此命令获取有关要移除的授权的信息。

## 管理远程会话

您可以使用 View PowerCLI cmdlet 管理远程会话。

**表 3-20. 常见远程会话管理任务示例**

任务	View PowerCLI cmdlet 语法示例
列出特定域用户的所有活动远程会话	Get-RemoteSession -username mydom\fred
断开特定域用户的所有活动会话连接	Get-RemoteSession -username mydom\fred   Send-SessionDisconnect

任务	View PowerCLI cmdlet 语法示例
注销特定域用户的所有活动远程会话	<code>Get-RemoteSession -username mydom\fred   Send-SessionLogoff</code>
注销使用 RDP 显示协议的所有活动远程会话	<code>Get-RemoteSession -protocol RDP   Send-SessionLogoff</code>

## 管理虚拟机

您可以使用 View PowerCLI cmdlet 管理虚拟机。Horizon Agent 必须在虚拟机中处于正在运行状态。

**表 3-21. 常见虚拟机管理任务示例**

任务	View PowerCLI cmdlet 语法示例
获取有关特定桌面池中虚拟机的信息	<code>Get-DesktopVM -pool_id dtpool-3</code>
获取有关在特定 vCenter Server 实例上配置的虚拟机的信息	<code>Get-DesktopVM -vc_id (Get-ViewVC -serverName vc03.local.int).vc_id</code>
获取由置备特定桌面池的同一 vCenter Server 实例管理的虚拟机的相关信息	<code>Get-ViewVC -pool_id dtpool-1   Get-DesktopVM</code>
获取由特定 vCenter Server 实例管理的所有虚拟机的相关信息	<code>Get-ViewVC -serverName vc01.mydom.int   Get-DesktopVM</code>
列出特定虚拟机的所有处于活动状态的永久用户数据磁盘	<code>Get-ProfileDisk -VMname vm01</code>
重置特定桌面池中的所有虚拟机	<code>Get-Pool -pool_id dtpool-05   Get-DesktopVM   Send-VMReset</code>
重置具有特定显示名称的桌面池的虚拟机	<code>Get-Pool -displayName dtp1   Get-DesktopVM   Send-VMReset</code>

## 显示有关物理机的信息

您可以使用 `Get-DesktopPhysicalMachine` cmdlet 显示有关物理机的信息。

在此示例中，`Get-DesktopPhysicalMachine` cmdlet 显示有关具有特定 IP 地址的物理机的信息。

```
Get-DesktopPhysicalMachine -hostname myhost01
```

## 更新虚拟机所有权

您可以使用 `Update-UserOwnership` 和 `Remove-UserOwnership` cmdlet 更新虚拟机的所有权。

在此示例中，`Update-UserOwnership` cmdlet 为用户 `user1` 更新虚拟机 `vm04` 的所有权。

```
Update-UserOwnership -machine_id (Get-DesktopVM -Name "vm04").machine_id
-sid (Get-User -name usr1).sid
```

在此示例中，`Remove-UserOwnership` cmdlet 移除虚拟机 `vm22` 的所有权。

```
Remove-UserOwnership -machine_id (Get-DesktopVM -Name "vm22").machine_id
```

## 显示事件报告

您可以使用 View PowerCLI cmdlet 显示事件报告。

**表 3-22. 常见事件报告任务示例**

任务	View PowerCLI cmdlet 语法示例
列出所有可用事件报告视图	<code>Get-EventReportList</code>
显示某个特定日期之后发生的所有配置更改事件	<code>Get-EventReport -viewName config_changes -startDate (Get-Date -Year 2011 -Month 5 -Day 20 -Hour 0 -Minute 0 -Second 0)</code>
显示两个特定日期之间发生的所有用户事件	<code>Get-EventReport -viewName user_events -startDate (Get-Date -Year 2011 -Month 12 -Day 1 -Hour 0 -Minute 0 -Second 0) -endDate (Get-Date -Year 2011 -Month 12 -Day 2 -Hour 0 -Minute 0 -Second 0)</code>
显示过去 24 个小时内发生的所有用户事件	<code>Get-EventReport -viewName user_events -startDate ((Get-Date).AddDays(-1))</code>
显示本年度发生的所有用户事件。	<code>Get-EventReport -viewName user_events -startDate (Get-Date -Day 01 -Month 01 -Hour 0 -Minute 0 -Second 0)</code>

## 显示和更新全局设置

您可以使用 View PowerCLI cmdlet 显示和更新 View 的全局设置。

**表 3-23. 常见全局设置管理任务示例**

任务	View PowerCLI cmdlet 语法示例
显示全局设置	<code>Get-GlobalSetting</code>
更新会话超时设置	<code>Update-GlobalSetting -SessionTimeout 1800</code>
更新强制注销警告消息和延迟时限	<code>Update-GlobalSetting -DisplayLogoffWarning \$true -ForcedLogoffAfter \$logoutdelay -ForcedLogoffMessage "Forced log out will occur in \$logoutdelay minutes"</code>
要求客户端使用 SSL 进行连接，并设置登录前消息	<code>Update-GlobalSetting -UseSSLClient \$true -PreLoginMessage "Insert disclaimer and other notices here."</code>

## 显示和添加许可证密钥

您可以使用 `Get-License` 和 `Set-License` cmdlet 显示和添加 View 的许可证密钥。

在此示例中，`Get-License` cmdlet 显示已安装的许可证密钥。

```
Get-License
```

在此示例中，`Set-License` cmdlet 添加一个许可证密钥。

```
Set-License -key "08A25-0212B-0212C-4D42E"
```



## 使用 View PowerCLI 执行高级任务的示例

您可以将 View PowerCLI 和 vSphere PowerCLI cmdlet 相结合，来创建可执行调整池大小和向桌面池添加数据存储等复杂操作的 PowerShell 函数。

### 确定 View 连接服务器是否正在运行

以下 PowerShell 函数确定 View 连接服务器服务是否正在运行，如果未运行，则启动该服务。

```
# WaitForViewStartup
# Parameters
# $ClearError If $true, clear the $error object on completion.
# $StartBroker If $true, start the service if it is not running.

function WaitForViewStartup
{ param ($ClearError = $true, $StartBroker = $true)
  $service = Get-Service wsbroker
  if($service -and (Get-Service wstomcat)){
    $started = $false
    if($service.Status -eq "Stopped"){
      if($StartBroker){ # Start the broker if it is not running.
        Write-Warning "Connection Broker service is stopped, attempting to start."
        $errCountBefore = $error.Count
        Start-Service wsbroker
        $errCountAfter = $error.Count
        if($errCountAfter -gt $errCountBefore){
          break
        }
      }
    } else {
      Write-Error "Connection Broker service is stopped."
      break
    }
  }
  while(!$started){ # Loop until service has completed starting up.
    Write-Warning "Waiting for View Connection Server to start."
    $errCountBefore = $error.Count
    $outPut = Get-GlobalSetting -ErrorAction SilentlyContinue
    $errCountAfter = $error.Count
    $started = $true
    if($errCountAfter -gt $errCountBefore){
      $err = $error[0].ToString()
      if($err.Contains("NoQueueHandler")){
        $started = $false
        Start-Sleep -s 1
      } else {
        if($ClearError){
          $error.Clear()
        }
        Write-Error $err
        break
      }
    }
  }
  if($ClearError){
    $error.Clear()
  }
}
```

```

    }
}
} else {
    Write-Error "The View Connection Server services could not be found. Is the Connection
        Server installed?"
}
}

```

## 调整自动和链接克隆池的大小

以下 PowerShell 函数确定所有桌面池的当前使用情况，并调整已达到最大容量的任何自动置备桌面池或链接克隆桌面池的大小。

```

# PollAllPoolsUsage
# Parameters
# $increment Amount by which to increase a pool that is at maximum capacity (default = 5).

function PollAllPoolsUsage
{ param ($increment)

    if(-not $increment){
        $increment = 5
    }
    # Retrieve all pool objects and check each one individually
    $pools = Get-Pool
    foreach ($pool in $pools){
        PollPoolUsage $pool $increment
    }
}

# PollPoolUsage
# Parameters
# $Pool Pool object that represents the pool to be checked.
# $increment Amount by which to increase pool that is at maximum capacity.

function PollPoolUsage
{ param ($Pool, $increment)

    # Get a list of remote sessions for the pool (errors are suppressed)
    $remotes = Get-RemoteSession -pool_id $Pool.pool_id -ErrorAction SilentlyContinue
    # Count the remote sessions.
    $remotecount = 0
    if($remotes){
        $remotecount = ([Object[]]($remotes)).Count
    }

    # Determine the maximum number of desktops configured for a pool.
    $maxdesktops = 0
    if($Pool.deliveryModel -eq "Provisioned"){
        $maxdesktops = $Pool.maximumCount
    } else {
        $maxdesktops = $Pool.machineDNs.split(";").Count
    }

    # Output the usage statistics for a pool.
    Write-Output ("==== " + $Pool.pool_id + " ====")
}

```

```

Write-Output ("Remote session count: " + $remotecount)
Write-Output ("Maximum desktops: " + $maxdesktops)
# If a pool is using all its desktops, increase its maximum size
# or output a warning if it cannot be resized.
if($maxdesktops -eq $remotecount){
    if($Pool.deliveryModel -eq "Provisioned"){ # Pool type can be resized
        $newmaximum = [int]$Pool.maximumCount + [int]$increment
        if($Pool.desktopSource -eq "VC"){ # Resize an automatic pool
            Update-AutomaticPool -pool_id $Pool.pool_id -maximumCount $newmaximum
        } elseif ($Pool.desktopSource -eq "SVI"){ # Resize a linked-clone pool
            Update-AutomaticLinkedClonePool -pool_id $Pool.pool_id -maximumCount $newmaximum
        }

        Write-Output ("Pool " + $Pool.pool_id + " is using 100% of its desktops. Maximum VMs
            increased to " + $newmaximum)
    } else { # Pool type cannot be resized
        Write-Output ("Pool " + $Pool.pool_id + " is using 100% of its desktops. Consider
            increasing its capacity.")
    }
}
}
}

```

## 确定 vSphere 清单对象的路径

以下 PowerShell 函数使用 vSphere PowerCLI 返回 vSphere 清单对象的完整路径。

```

# VVGetInventoryPath
# Parameters
# $InvObject Inventory object in vSphere PowerCLI.
#
# Examples
# VVGetInventoryPath (Get-VM -name myVM)
# VVGetInventoryPath (Get-ResourcePool | Select -first 1)

function VVGetPath($InvObject){
    if($InvObject){

        $ObjectType = $InvObject.GetType().Name
        $ObjectBaseType = $InvObject.GetType().BaseType.Name
        if($ObjectType.Contains("DatastoreImpl")){
            Write-Error "Use the VVGetDataStorePath function to determine datastore paths."
            break
        }
        if(-not ($ObjectBaseType.Contains("InventoryItemImpl") -or
            $ObjectBaseType.Contains("FolderImpl") -or
            $ObjectBaseType.Contains("DatacenterImpl") -or
            $ObjectBaseType.Contains("VMHostImpl") ) ){
            Write-Error ("The provided object is not an expected vSphere object type. Object type
                is " + $ObjectType)
            break
        }
    }

    $path = ""
    # Recursively move up through the inventory hierarchy by parent or folder.

```

```

    if($InvObject.ParentId){
        $path = VVGetPath(Get-Inventory -Id $InvObject.ParentId)
    } elseif ($InvObject.FolderId){
        $path = VVGetPath(Get-Folder -Id $InvObject.FolderId)
    }

    # Build the path, omitting the "Datacenters" folder at the root.
    if(-not $InvObject.isChildTypeDatacenter){ # Add object to the path.
        $path = $path + "/" + $InvObject.Name
    }
    $path
}
}

```

## 确定 vSphere 数据存储对象的路径

以下 PowerShell 函数按照资源池指定的方式使用 vSphere PowerCLI 返回数据存储在群集中的完整路径。

```

# VVGetDatastorePath
# Parameters
#     $Datastore Datastore object in vSphere PowerCLI.
#     $ResourcePool Resource pool in cluster.
#
#Example
#           VVGetDatastorePath (Get-Datastore "datastore1") (Get-ResourcePool "Resources")

function VVGetDatastorePath($Datastore,$ResourcePool){
    if($Datastore -and $ResourcePool){

        $dsType = $Datastore.GetType().Name
        $rpType = $ResourcePool.GetType().Name
        if(-not ($dsType.Contains("Datastore"))) ){
            Write-Error "The Datastore provided is not a Datastore object."
            break
        }
        if(-not ($rpType.Contains("ResourcePool"))) ){
            Write-Error "The Resource Pool provided is not a ResourcePool object."
            break
        }

        $ClusterPath = VVGetPath(Get-Inventory -Id $ResourcePool.ParentId)
        $path = $ClusterPath + "/" + $Datastore.Name
        $path
    }
}

```

## 添加和移除数据存储

您可以定义 PowerShell 函数来添加和移除数据存储。

下例中的 PowerShell 函数可为自动池添加和移除数据存储。

```

# AddDatastoreToAutomaticPool
# Parameters

```

```
#          $Pool Pool ID of pool to be updated.
#          $Datastore Full path to datastore to be added.

function AddDatastoreToAutomaticPool
{ param ($Pool, $Datastore)
    $PoolSettings = (Get-Pool -pool_id $Pool)
    $datastores = $PoolSettings.datastorePaths + ";$Datastore"
    Update-AutomaticPool -pool_id $Pool -datastorePaths $datastores
}
Define a PowerShell function to remove a datastore from an automatic pool.

# RemoveDatastoreFromAutomaticPool
# Parameters
#   $Pool Pool ID of pool to be updated.
#   $Datastore Full path to datastore to be removed.

function RemoveDatastoreFromAutomaticPool
{ param ($Pool, $Datastore)
    $PoolSettings = (Get-Pool -pool_id $Pool)
    $currentdatastores = $PoolSettings.datastorePaths

    $datastores = ""
    foreach ($path in $currentdatastores.split(";")){
        if(-not ($path -eq $Datastore)){
            $datastores = $datastores + "$path;"
        }
    }
    Update-AutomaticPool -pool_id $Pool -datastorePaths $datastores
}
```

下例中的 PowerShell 函数可为链接克隆池添加和移除数据存储。

```
# AddDatastoreToLinkedClonePool
# Parameters
#   $Pool Pool ID of pool to be updated.
#   $Datastore Full path to datastore to be added.

function AddDatastoreToLinkedClonePool
{ param ($Pool, $Datastore)
    $PoolSettings = (Get-Pool -pool_id $Pool)
    $datastores = $PoolSettings.datastoreSpecs + ";$Datastore"
    Update-AutomaticLinkedClonePool -pool_id $Pool -datastoreSpecs $datastores
}

Define a PowerShell function to remove a datastore from a linked-clone pool.

# RemoveDatastoreFromLinkedClonePool
# Parameters
#   $Pool Pool ID of pool to be updated.
#   $Datastore Full path to datastore to be removed.

function RemoveDatastoreFromLinkedClonePool
{ param ($Pool, $Datastore)
    $PoolSettings = (Get-Pool -pool_id $Pool)
    $currentdatastores = $PoolSettings.datastoreSpecs
```

```

$datastores = ""
foreach ($spec in $currentdatastores.split(";")){
    $path = $spec.split("[1]")[1]
    $pathToRemove = $Datastore.split("[1]")[1]
    if(-not $pathToRemove){
        $pathToRemove = $Datastore
    }
    if(-not ($path -eq $pathToRemove)){
        $datastores = $datastores + "$spec;"
    }
}
Update-AutomaticLinkedClonePool -pool_id $Pool -datastoreSpecs $datastores
}

```

## 向桌面池分配多个网络标签

在 View 5.2 和更高版本中，您可以将自动的完整克隆和链接克隆桌面池配置为使用多个网络标签。此功能扩展了可分配到池中虚拟机的 IP 地址数量，从而可以更轻松地创建具有大量桌面的池。您可以使用 **View PowerCLI cmdlet** 分配可在部署了自动完整克隆或链接克隆桌面池的 **vCenter Server** 资源池中使用的网络标签。

默认情况下，桌面池中的虚拟机从父虚拟机或模板继承网络接口卡 (**Network Interface Card, NIC**) 及其关联的网络标签。某些父虚拟机或模板可能具有多个网卡和关联的网络标签。通常，由网络标签定义的 **VLAN** 子网掩码可提供数量有限的可用 IP 地址。例如，子网掩码最多可以提供 **254** 个可分配给桌面虚拟机的 IP 地址。

**View** 在整个桌面池中的虚拟机之间分发网络标签。当 **View** 置备桌面时，它将按字母顺序分配网络标签。当通过使用第一个网络标签的 IP 地址置备的虚拟机达到最大数量时，**View** 开始分配第二个标签，以此类推。

要将桌面池配置为使用多个网络标签，请从在 **vCenter Server** 中为 **ESXi** 群集定义的标签中选择网络标签、将标签与从父虚拟机或模板继承而来的网卡相关联、指定每个网络标签中可分配给虚拟机的最大 IP 地址数量，然后在网络标签配置文件中保存该信息。您可以在用来创建桌面池的 **View PowerCLI cmdlet** 中指定网络标签配置文件。

## 网络标签配置文件格式

您可以使用网络标签配置文件来将自动完整克隆和链接克隆桌面池配置为使用多个网络标签。网络标签配置文件中包含用于控制是否分配网络标签的标记和用于定义网卡、网络标签和网络标签属性的部分。

### Enabled 标记

默认情况下，此标记设置为 **enabled=true**。请将此标记保持设置为 **true** 以允许 **View** 向池分配网络标签。

### 网卡参数定义部分

此部分列出模板或父虚拟机快照上的 **vCenter Server** 中所定义的网卡。请勿编辑此部分。

## 网络参数定义部分

此部分列出在 vCenter Server 实例中为群集中的 ESXi 主机定义的网络标签。网络标签按字母顺序列出。如果群集使用标准的网络标签和分布式虚拟交换机网络标签，则针对一个池时仅应使用一种类型的标签。请勿编辑此部分。

## 网络标签属性定义部分

此部分列出与每个网卡关联的网络标签。网络标签已被注释掉，分配无效。您必须从相应的网络标签移除注释（### 标记），才能使它们可以分配给桌面池。

`maxvm` 参数用于定义网络标签中可分配到虚拟机的最大 IP 地址数量。`maxvm` 参数的值由 `Export-NetworkLabelSpecForLinkedClone` 或 `Export-NetworkLabelSpecForFullClone cmdlet` 中的 `-maxVMsPerNetworkLabel` 参数生成。您可以在配置文件中手动编辑此值。

最佳做法是，不要将一个网络标签分配给多个桌面池。最大网络标签数仅支持按每个网卡或每个池进行计算。例如，如果您将池 `pool1` 上的网卡 `NIC1` 配置为使用网络标签 `network06`，`maxvm` 为 `244`，然后您又将池 `pool2` 上的网卡 `NIC1` 配置为使用同一网络标签，即 `network06`，而 `maxvm` 也是 `244`，那么 `network06` 必须具有一个至少有 `488` 个地址的实际可分配 IP 地址空间，否则从该网络标签分配的 IP 可能会超额。

如果父虚拟机或模板具有两个网卡，则 `Export-NetworkLabelSpecForLinkedClone` 或 `Export-NetworkLabelSpecForFullClone cmdlet` 发现的每个网络标签均会与这两个网卡相关联。警告消息会说明，IP 地址分配可能会发生超额，因为网卡的分配函数无法知晓从同一个网络标签为第二个网卡进行的分配。每个网卡仅知晓自身的网络标签分配。

## 网络标签配置文件示例

以下示例显示了一个典型的网络标签配置文件。

```
#Network Label Configuration Spec
#WARNING! Setting enabled flag to false will
#turn off the automatic network label assignment
#for newly provisioned desktops.
enabled=true

#Parameter Definition for NIC
nic1=Network adapter 1
nic2=Network adapter 2

#Parameter Definition for Network
network01=dv_2k_2004
network02=dv_2kclient_2164
network03=dv_2kdt1_2084
network04=dv_2kInfra_1924
network05=dv_vMotion
network06=desktop-auto01-230
network07=desktop-auto02-240
network08=desktop-auto03-250
network09=desktop-auto04-260
network10=desktop-auto05-270

#Network Label Attribute Definition
```

```
#Expected format:
#<nic_param>.<network_param>.maxvm=<max vm for network label>

#WARNING! Multiple NICs:(nic1,nic2) detected against network01,
#total port count against network01: 488.
#IP address might be over-subscribed.
#WARNING! Multiple NICs:(nic1,nic2) detected against network02,
#total port count against network02: 488.
#IP address might be over-subscribed.
...
...
#WARNING! Multiple NICs:(nic1,nic2) detected against network02,
#total port count against network10: 488.
#IP address might be over-subscribed.

####nic1.network01.maxvm=244
####nic1.network02.maxvm=244
####nic1.network03.maxvm=244
####nic1.network04.maxvm=244
####nic1.network05.maxvm=244
####nic1.network06.maxvm=244
####nic1.network07.maxvm=244
####nic1.network08.maxvm=244
####nic1.network09.maxvm=244
####nic1.network10.maxvm=244
####nic2.network01.maxvm=244
####nic2.network02.maxvm=244
####nic2.network03.maxvm=244
####nic2.network04.maxvm=244
####nic2.network05.maxvm=244
####nic2.network06.maxvm=244
####nic2.network07.maxvm=244
####nic2.network08.maxvm=244
####nic2.network09.maxvm=244
####nic2.network10.maxvm=244
```

## 获取并导出网卡和网络标签信息

您必须先获取网卡和网络标签信息，并将该信息导出到网络标签配置文件，然后才能将多个网络标签分配给池。

对于完整克隆池，应获取有关在用于创建完整克隆池的模板上配置的网卡的信息。对于链接克隆池，应获取有关在用于创建链接克隆池的父虚拟机上配置的网卡的信息。网络标签信息可从在群集（要部署桌面池）的 ESXi 主机上配置的可用网络标签中获取。

在将网卡和网络标签信息导出到网络标签配置文件时，请设置每个网络标签中的 IP 地址可置备的最大虚拟机数量。应允许特定数量的虚拟机用于开销。例如，如果网络标签 VLAN 允许将最多 254 个 IP 地址分配给虚拟机，您可以将最大数量设置为 244。

### 前提条件

熟悉使用 View PowerCLI。请参阅 [View PowerCLI 使用入门](#)。



## 步骤

- ◆ 要获取并导出网卡和网络标签信息以创建完整克隆池，请使用 `Export-NetworkLabelSpecForFullClone` cmdlet。

例如：

```
Export-NetworkLabelSpecForFullClone -vc_id id -clusterPath "cluster-path"
-TemplatePath "template-path" -maxVMsPerNetworkLabel nn
-networkLabelConfigFile "config-file-path"
```

- ◆ 要获取并导出网卡和网络标签信息以创建链接克隆池，请使用 `Export-NetworkLabelSpecForLinkedClone` cmdlet。

例如：

```
Export-NetworkLabelSpecForLinkedClone -vc_id id -clusterPath "cluster-path"
-parentVMPATH "parent-vm-path" -parentSnapshotPath "snapshot-path"
-maxVMsPerNetworkLabel nn -networkLabelConfigFile "config-file-path"
```

网络标签 cmdlet 可获取在 vCenter Server 中为指定群集中的所有 ESXi 主机配置的网络标签，设置每个网络标签中的 IP 地址可置备的最大虚拟机数量，并将网卡和网络标签列表导出到网络标签配置文件。

## 示例：获取网卡和网络标签信息

在此示例中，`Export-NetworkLabelSpecForLinkedClone` cmdlet 从 Win7 父虚拟机和 Snapshot1 获取网卡，并从 Cluster1 获取网络标签信息。每个网络标签中的 IP 地址最多可用于置备 244 个虚拟机。此信息将导出到一个名为 `C:/label.txt` 的网络标签配置文件。

```
Export-NetworkLabelSpecForLinkedClone -vc_id 1a2b3c4d-5e6f
-clusterPath "/myresourcepool/host/Cluster1/"
-parentVMPATH "/myresourcepool/vm/Win7-Parent" -parentSnapshotPath "/snapshot1"
-maxVMsPerNetworkLabel 244 -networkLabelConfigFile "C:/label.txt"
```

## 后续步骤

验证并编辑网络标签配置文件。请参阅[验证并编辑网络标签配置文件](#)。

## 验证并编辑网络标签配置文件

生成网络标签配置文件后，您必须验证其内容并进行编辑以指定要分配给池的网卡和网络标签配对。

### 前提条件

- 生成网络标签配置文件。请参阅[获取并导出网卡和网络标签信息](#)。
- 熟悉网络标签配置文件格式。请参阅[网络标签配置文件格式](#)。

## 步骤

- 1 在文本编辑器中，打开网络标签配置文件。
- 2 对于要分配给池的每个网卡和网络标签配对，请移除注释（### 标记）。

- 3 确认每个网络标签均只分配给一个网卡。
- 4 确认池只使用一种网络标签类型。

网卡可配置为使用标准网络交换机或分布式虚拟网络交换机。

- 5 保存更改。

### 示例：已编辑的网络标签配置文件

在此示例中，network06 和 network07 分配给 nic1（网络适配器 1），network08 和 network09 分配给 nic2（网络适配器 2）。network01 到 network05 保持为注释，未被使用。此配置可支持最多包含 488 个虚拟机的池。

```
#Network Label Configuration Spec
...

#Network Label Attribute Definition
#Expected format:
#<nic_param>.<network_param>.maxvm=<max vm for network label>
...

####nic1.network01.maxvm=244
####nic1.network02.maxvm=244
####nic1.network03.maxvm=244
####nic1.network04.maxvm=244
####nic1.network05.maxvm=244
nic1.network06.maxvm=244
nic1.network07.maxvm=244
####nic1.network08.maxvm=244
####nic1.network09.maxvm=244
####nic1.network10.maxvm=244
####nic2.network01.maxvm=244
####nic2.network02.maxvm=244
####nic2.network03.maxvm=244
####nic2.network04.maxvm=244
####nic2.network05.maxvm=244
####nic2.network06.maxvm=244
####nic2.network07.maxvm=244
nic2.network08.maxvm=244
nic2.network09.maxvm=244
####nic2.network10.maxvm=244
```

### 后续步骤

创建可使用多个网络标签的桌面池。请参阅[部署使用多个网络标签的桌面池](#)。

## 部署使用多个网络标签的桌面池

您可以通过 View PowerCLI cmdlet 部署使用多个网络标签的自动完整克隆或链接克隆桌面池。

### 前提条件

- 生成网络标签配置文件。请参阅[获取并导出网卡和网络标签信息](#)。

- 验证并编辑网络标签配置文件。请参阅[验证并编辑网络标签配置文件](#)。

## 步骤

- ◆ 使用 View PowerCLI cmdlet 部署桌面池。

选项	操作
部署链接克隆池	运行 Add-AutomaticLinkedClonePool cmdlet，并使用 -NetworkLabelConfigFile 参数指定网络标签配置文件的路径。
部署完整克隆桌面池	运行 Add-AutomaticPool cmdlet，并使用 -NetworkLabelConfigFile 参数指定网络标签配置文件的路径。

## 示例：链接克隆桌面池部署

在此示例中，Add-AutomaticLinkedClonePool cmdlet 创建名为 POOL\_NAME 的链接克隆池，并在名为 label.txt 的网络标签配置文件中分配网络标签。

```
Add-AutomaticLinkedClonePool -Pool_id POOL_NAME -NamePrefix NAME_PREFIX -Vc_id
2162aa44-e99c-4f1a-875d-dd295681d2ca -Persistence Persistent
-VmFolderPath "/resourcepool/vm/Discovered virtual machine"
-ResourcePoolPath "/resourcepool/host/Cluster1/Resources/Cluster1_pool1"
-ParentVmPath "/resourcepool/vm/Win7-Parent" -ParentSnapshotPath
"/780936_agent/noServiceRestart/updatedVPListener" -DatastoreSpecs
"[Conservative,replica]/resourcepool/host/Cluster1/Cluster1_Replica1;[Moderate,OS,data]/resourcepool/host/Cluster1/Desktop_FC_9;[Moderate,OS,data]/resourcepool/host/Cluster1/Desktop_FC_8;[Moderate,OS,data]/resourcepool/host/Cluster1/Desktop_FC_7" -Composer_ad_id
1a2b3c4d-5d6e-7f-1234-1234abcd -UseUserDataDisk $false -UseTempDisk
$false -MinimumCount 2000 -MaximumCount 2000 -HeadroomCount 1500
-PowerPolicy AlwaysOn -SuspendProvisioningOnError $false
-NetworkLabelConfigFile "C:/label.txt"
```

## 分配网络标签

在 View 置备池中的虚拟机时，它会将网络标签分配给虚拟机中的网卡。当某个特定网络标签可以分配的虚拟机达到最大数量后，View 开始从下一个网络标签进行置备。

View 将网络标签分配给虚拟机后，不会更改该分配。

如果删除池中的虚拟机，则会释放关联的网络标签分配，然后可以将其分配给池中的任何新置备虚拟机。

View Composer 刷新操作不会影响网络标签分配。虚拟机会继续接收首次置备时分配给它们的网络标签。

## 在重构和重新平衡操作期间保留网络标签

在 View Composer 重构和重新平衡操作期间，View 尝试确保在链接克隆从新基础映像继承新网卡时，保留连接到每个链接克隆桌面的每个网卡的网络标签。只要新基础映像的可用网卡配置了相同类型的网络交换机，View 就会保留在执行重构和重新平衡操作之前已有网卡的网络标签。您可以使用标准网络交换机或分布式虚拟网络交换机配置网卡。

以下示例介绍在重构和重新平衡操作期间用来控制网络标签保留的规则。

- 如果原始链接克隆桌面和新基础映像均具有一个配置了标准网络交换机的网卡，则将保留网络标签和 MAC 地址。
- 如果原始链接克隆桌面和新基础映像均具有一个配置了分布式虚拟网络交换机的网卡，则将保留网络标签和 MAC 地址。
- 如果原始链接克隆桌面具有一个配置了标准网络交换机的网卡，而新基础映像具有一个配置了分布式虚拟网络交换机的网卡，则不保留网络标签，但会保留 MAC 地址。
- 如果原始链接克隆桌面具有两个配置了标准网络交换机的网卡，新基础映像具有一个配置了标准网络交换机的网卡，将保留一个原始网卡的网络标签，但不会保留与另一原始网卡关联的所有网络属性。

## 显示池的网络标签分配

您可以使用 `Get-Pool cmdlet` 显示池的网络标签分配。`networkLabelSpecs` 输出参数显示分配给该池的网络标签。您无需在命令行中添加参数即可显示此信息。

下表介绍了 `networkLabelSpecs` 输出格式。

**表 3-24. networkLabelSpecs 输出格式**

值	说明
nl	网络标签名称。
nic	网卡名称。
enabled	值为 1 表示标签处于活动状态。值为 0 表示标签已被禁用。
max	可分配给网络标签的最大虚拟机数量。
usage	当前已分配给网络标签的虚拟机数量。 当 <code>usage</code> 值等于 <code>max</code> 值时，View 会停止分配该网络标签，并从下一个可用的标签开始进行分配。

以下 `Get-Pool cmdlet` 示例显示池 `Pool2` 的网络标签分配。

```
> get-pool -pool_id Pool2
...
networkLabelSpecs : [nl=desktop-auto01-230;nic=Network adapter 1;enable
d=1;max=239;usage=239];[nl=desktop-auto02-240;nic=N
etwork adapter 1;enabled=1;max=239;usage=239];[nl=vie
wscale-auto03-250;nic=Network adapter 1;enabled=1;max
=239;usage=239];[nl=desktop-auto04-260;nic=Network
adapter 1;enabled=1;max=239;usage=239];[nl=desktopauto05-
270;nic=Network adapter 1;enabled=1;max=239;us
age=239];[nl=desktop-auto06-280;nic=Network adapter
1;enabled=1;max=239;usage=239];[nl=desktop-auto07-
290;nic=Network adapter 1;enabled=1;max=239;usage=239
];[nl=desktop-auto08-300;nic=Network adapter 1;enab
led=1;max=239;usage=239];[nl=desktop-auto09-310;nic
=Network adapter 1;enabled=1;max=239;usage=88] ...
```

## 显示虚拟机的网络标签分配

您可以使用 `Get-DesktopVM cmdlet` 显示虚拟机的网络标签分配。`netLabelAutoAssigns` 输出参数显示 View 尝试分配给虚拟机的网卡名称和网络标签。您无需在命令行中添加参数即可显示此信息。

如果网络标签配置文件中的 `enabled` 标记设置为 0，`netLabelAutoAssigns` 输出会显示此功能已被禁用。

`netLabelAutoAssigns` 输出显示 View 连接服务器为虚拟机预留的网络标签分配。要查看在 vCenter Server 中为虚拟机分配的网络标签，请在 `Get-DesktopVM cmdlet` 中使用 `-getNetworkLabel` 参数。有关更多信息，请参阅[显示虚拟机的 vCenter Server 网络标签分配](#)。

以下 `Get-DesktopVM cmdlet` 示例显示池 `pool2` 中的虚拟机 918 的网络标签分配。

```
> get-desktopvm -pool_id pool2
...
vm : 918
ps_object_type : vc_vm
id : VirtualMachine-vm-47878
vc_id : 2162aa44-e99c-4f1a-875d-dd295681d2ca
Name : pool2-1811
UnescapedName : pool2-1811
Path : /resource/vm/Discovered virtual machine/pool2/pool2-1811
GuestFullName : Microsoft Windows 7 (32-bit)
GuestID : windows7Guest
HostName : POOL2-1811.vdi3.net
IPAddress : 10.143.30.205
machine_id : 56496104-bf91-4d69-8bae-fb4493608542
user_sid :
user_displayname :
isInPool : true
pool_id : pool2
isLinkedClone : true
composerTask : refresh
netLabelAutoAssigns : {Network adapter 1=desktop-auto08-300} ...
```

## 显示虚拟机的 vCenter Server 网络标签分配

vCenter Server 可以为虚拟机分配其他网络标签。这些分配不在 View 的控制范围内。

要查看在 vCenter Server 中为虚拟机分配的网络标签，请在 `Get-DesktopVM cmdlet` 中使用 `-getNetworkLabel` 参数。您必须在命令行中键入布尔值 `$true`，以启用 `-getNetworkLabel` 参数。`Get-DesktopVM cmdlet` 的输出显示 `networkLabels` 参数，它会显示为虚拟机进行的网卡和网络标签分配。

以下 `Get-DesktopVM cmdlet` 示例显示在 vCenter Server 中为池 `pool2` 中的虚拟机 1849 进行的网络标签分配。

```
> get-desktopvm -pool_id pool2 -getnetworklabel $true
...
vm : 1849
ps_object_type : vc_vm
id : VirtualMachine-vm-46148
vc_id : 2162aa44-e99c-4f1a-875d-dd295681d2ca
```

```

Name : pool2-85
UnescapedName : pool2-85
Path : /resource/vm/Discovered virtual machine/pool2/pool2-85
GuestFullName : Microsoft Windows 7 (32-bit)
GuestID : windows7Guest
HostName : P00L2-85.vdi3.net
IPAddress : 192.168.1.10
networkLabels : {Network adapter 1=desktop-auto01-230}
machine_id : be14deda-ec1b-4dd1-834a-915fcc7d51a0
user_sid :
user_displayname :
isInPool : true
pool_id : pool2
isLinkedClone : true
composerTask :
netLabelAutoAssigns : {Network adapter 1=desktop-auto01-230} ...

```

---

**注** 由于 `-getNetworkLabel` 参数的运行时间较长，请在 vSphere 的非使用高峰时段运行带有 `-getNetworkLabel` 参数的 `Get-DesktopVM` cmdlet。

---

## 禁用自动网络标签分配

您可以在使用自动分配的现有池上禁用网络标签分配。

### 步骤

- 1 在文本编辑器中，打开网络标签配置文件。
- 2 将 `enabled` 标记设置为 `false`。  
例如: `enabled=false`
- 3 保存更改。
- 4 运行 `Update-AutomaticLinkedClonePool` 或 `Update-AutomaticPool` cmdlet，并使用 `-NetworkLabelConfigFile` 参数指定更新网络标签配置文件的路径。

当 View 置备池中的新虚拟机时，将使用父虚拟机或模板上的网络标签。

# 自定义 LDAP 数据

您可以使用 VMware 和 Microsoft 命令行工具将 LDAP 配置数据导入到 View，或从其中导出。这些命令行工具以 LDAP 数据交换格式 (LDAP Data Interchange Format, LDIF) 配置文件导入和导出 LDAP 配置数据。

此功能专供要执行自动批量配置操作的高级管理员使用。要创建用于更新 View 配置的脚本，请使用 View PowerCLI。

本章讨论了以下主题：

- [LDAP 配置数据简介](#)
- [修改 LDAP 配置数据](#)

## LDAP 配置数据简介

所有 View 配置数据均存储在 LDAP 目录中。每个 View 连接服务器的标准实例或副本实例都包含一个本地 LDAP 配置存储库及与每个 View 连接服务器实例之间的复制协议。这样可确保对一个存储库的更改会自动复制到其他所有存储库。

当您使用 View Administrator 修改 View 配置时，存储库中的相应 LDAP 数据也会随之更新。例如，在您添加桌面池时，View 会将与用户、用户组和授权相关的信息存储在 LDAP 中。View 连接服务器实例可自动管理其他 LDAP 配置数据，并使用存储库中的信息控制 View 操作。

您可以使用 LDIF 配置文件执行多项任务，包括在 View 连接服务器实例之间传输配置数据，以及备份 View 配置以便您可以还原 View 连接服务器实例的状态。

您还可以使用 LDIF 配置文件定义大量 View 对象（如桌面池），并将这些对象添加到 View 连接服务器实例，而无需使用 View Administrator 来手动执行此任务。

在 View 3.1 和更高版本中，View 会定期执行 LDAP 存储库备份。

LDAP 配置数据以 ASCII 纯文本形式传输，并遵循 Internet 工程任务组 (Internet Engineering Task Force, IETF) RFC 2849 标准。

## 修改 LDAP 配置数据

您可以将 View 连接服务器实例上的 LDAP 配置数据导出到 LDIF 配置文件，修改 LDIF 配置文件，并将修改的 LDIF 配置文件导入到其他 View 连接服务器实例，以执行自动批量配置操作。

通过查看导出的 LDIF 配置文件的内容，您可以获取 View 中任何 LDAP 配置数据项的 LDIF 语法示例。例如，可以提取桌面池的数据，并以该数据作为模板来创建大量的桌面池。

## 导出 LDAP 配置数据

您可以使用 `vdmexport` 命令行实用程序将配置数据从标准或副本 View 连接服务器实例导出到 LDIF 配置文件。

默认情况下，`vdmexport` 命令行实用程序安装在 `C:\Program Files\VMware\VMware View\Server\tools\bin` 目录中。

### 步骤

- 1 登录标准或副本 View 连接服务器实例。

选项	操作
View 3.1 和更低版本	以管理员身份登录并成为本地管理员用户组的成员。
View 4.5 和更高版本	以角色为管理员或管理员（只读）的用户身份进行登录。
	<b>注</b> 您必须以角色为管理员或管理员（只读）的用户身份登录，以从 View 配置存储库导出配置数据。

- 2 在命令提示符中，键入 `vdmexport` 命令，并使用 `-f` 选项指定要导出的 LDIF 配置文件的名称。

例如：`vdmexport -f myexport.LDF`

或者，您也可以将输出重定向，而不使用 `-f` 选项。

例如：`vdmexport > myexport.LDF`

`vdmexport` 命令将 View 连接服务器实例的配置写入到您指定的文件中。如果您的角色没有足够的特权来查看配置存储库中的数据，则该命令将显示错误。

## 在 LDIF 配置文件中定义桌面池

您可以在 LDIF 配置文件中定义桌面池，然后导入自定义的 LDIF 配置文件，以创建大量的桌面池。

**注** 您还可以为在 LDAP 存储库中定义的其他对象创建自定义的 LDIF 配置文件，其中包括全局配置设置、特定 View 连接服务器实例或安全服务器的配置设置以及特定用户的配置设置。

要在 LDIF 配置文件中定义桌面池，您必须将以下条目添加到该文件。

- 桌面池中每个虚拟桌面的虚拟桌面虚拟机条目
- 每个桌面池的虚拟机池条目
- 定义桌面池授权的桌面应用程序条目

以一对一的关系，将每个虚拟机池条目与一个桌面应用程序条目相关联。桌面应用程序条目不能在虚拟机池条目之间共享，而一个虚拟机池条目也只能与一个桌面应用程序条目相关联。

下表介绍了修改 LDIF 配置文件中的桌面池定义时必须指定的属性。



表 4-1. 用于定义桌面池的重要属性

条目	属性	说明
虚拟桌面虚拟机 虚拟机池 桌面应用程序	cn	条目的公用名称。如果需要自动生成名称，请指定全局唯一标识符 (Globally Unique Identifier, GUID) 字符串。您可以使用任何可靠的 GUID 生成器，如 .NET 提供的机制（例如，通过调用 Visual Basic 中的 System.Guid.NewGuid().ToString()）。
桌面应用程序	成员	<p>有权访问桌面池的 Active Directory (AD) 用户和组的列表。该属性以 Windows 安全标识符 (Security Identifier, SID) 引用的形式指定。成员值 &lt;SID=S-1-2-3-4&gt; 表示 SID 值为 S-1-2-3-4 的 AD 用户或组。</p> <p>在 LDIF 格式中，左尖括号 (&lt;) 是保留的字符，因此您必须在属性名称后加上两个冒号 (::)，并使用 base 64 格式指定 SID 值（例如，PFNJRD1TLTetMi0zLTQ+IA==）。由于此属性有多个值，您可以在多个行上使用它来表示 SID 列表中的每个条目。</p>

## 示例 LDIF 配置文件桌面池项

以下示例摘录了 LDIF 配置文件中的部分内容。它显示名为 Pool1 的桌面池的示例项，该桌面池包含两个名称分别为 VM1 和 VM2 的虚拟桌面。该桌面池项与同样名为 Pool1 的桌面应用程序条目配对。

```
#
# Virtual Desktop VM entry VM1
#
DN: CN=vm1,OU=Servers,DC=vdi,DC=vmware,DC=int
changetype: add
objectClass: top
objectClass: pae-Server
objectClass: pae-WinServer
objectClass: pae-ThinWinServer
objectClass: pae-VM
cn: vm1
description: sample virtual desktop entry
pae-VmSuspended:: IA==
pae-OptIgnoreProcessList: 0
pae-MOID: vm-1
pae-VmState: READY
pae-ServerManaged: 1
pae-SSOEnabled: 1
pae-DisplayName: virtual desktop 1
pae-TunneledConnection: 1
pae-pwdEncryption: KERBS
ipHostNumber: vm1
pae-ClientProtVersion: 1
pae-WinDomain: NULL
pae-thinProto: XP_RDP
pae-Services: SESSION |, HEARTBEAT |, EVENTS |, USED |
pae-VmPath: /New Datacenter/vm/vm-1
pae-OptSuspendTimeout: 0
pae-OptDisconnectLimitTimeout: 0
pae-OptMaximumSessions: 0
pae-Disabled: 0

#
```

```

# Virtual Desktop VM entry VM2
#
DN: CN=vm2,OU=Servers,DC=vdi,DC=vmware,DC=int
changetype: add
objectClass: top
objectClass: pae-Server
objectClass: pae-WinServer
objectClass: pae-ThinWinServer
objectClass: pae-VM
cn: vm2
description: sample virtual desktop entry
pae-VmSuspended:: IA==
pae-OptIgnoreProcessList: 0
pae-MOID: vm-2
pae-VmState: READY
pae-ServerManaged: 1
pae-SSOEnabled: 1
pae-DisplayName: virtual desktop 2
pae-TunneledConnection: 1
pae-pwdEncryption: KERB5
ipHostNumber: vm2
pae-ClientProtVersion: 1
pae-WinDomain: NULL
pae-thinProto: XP_RDP
pae-Services: SESSION |, HEARTBEAT |, EVENTS |, USED |
pae-VmPath: /New Datacenter/vm/vm-2
pae-OptSuspendTimeout: 0
pae-OptDisconnectLimitTimeout: 0
pae-OptMaximumSessions: 0
pae-Disabled: 0
#
# Further Virtual Desktop VM entries as required
#
#
# VM Pool entry Pool1
#
DN: CN=Pool1,OU=Server Groups,DC=vdi,DC=vmware,DC=int
changetype: add
objectClass: top
objectClass: pae-ServerPool
cn: Pool1
pae-VCDN: CN=b180b93b-2dd3-4b58-8a81-b8534a4b7565,OU=VirtualCenter,OU=Properties,DC=vdi,
DC=vmware,DC=int
pae-MemberDN: CN=vm1,OU=Servers,DC=vdi,DC=vmware,DC=int
pae-MemberDN: CN=vm2,OU=Servers,DC=vdi,DC=vmware,DC=int
pae-VmPowerPolicy: remainon
pae-VmProvEnabled: 1
pae-VmProvSuspendOnError: 1
pae-VmStartClone: 1
pae-VmPoolCalculatedValues: 1
pae-ServerPoolType: 0
pae-VmMinimumCount: 0
pae-VmHeadroomCount: 0
pae-VmMaximumCount: 0
pae-Disabled: 0

```

```
#
# Desktop Application entry Pool1 -- one entry is required for each VM Pool
#
DN: CN=Pool1,OU=Applications,DC=vdi,DC=vmware,DC=int
changetype: add
objectClass: top
objectClass: pae-Entity
objectClass: pae-App
objectClass: pae-WinApp
objectClass: pae-ThinWinApp
objectClass: pae-DesktopApplication
cn: Pool1
member:: PFNJRD1TLTEtMi0zLTQ+IA==
pae-Icon: /thinapp/icons/desktop.gif
pae-URL: \
pae-Servers: CN=Pool1,OU=Server Groups,DC=vdi,DC=vmware,DC=int
pae-ServerProtocolLevel: OSX_NETOP
pae-ServerProtocolLevel: OS2_NETOP
pae-ServerProtocolLevel: NT4_NETOP
pae-ServerProtocolLevel: WIN2K_NETOP
pae-ServerProtocolLevel: NT4_RDP
pae-ServerProtocolLevel: WIN2K_RDP
pae-ServerProtocolLevel: XP_RDP
pae-Disabled: 0
```

## 使用 vdmimport 命令导入 LDAP 配置数据

在 View 4.5 和更高版本中，您可以使用 `vdmimport` 命令将配置数据从 LDIF 配置文件导入标准或副本 View 连接服务器实例。

默认情况下，`vdmimport` 命令行实用程序安装在 `C:\Program Files\VMware\VMware View\Server\tools\bin` 目录中。

### 前提条件

- 确认您使用的是 View 4.5 或更高版本。如果使用的是较低的 View 版本，请参阅[使用 LDIFDE 命令导入 LDAP 配置数据](#)。
- 将 LDAP 配置数据导出到 LDIF 配置文件。请参阅[导出 LDAP 配置数据](#)。

### 步骤

- 1 以角色为管理员的用户身份登录到 View 连接服务器实例。

您必须以角色为管理员的用户身份登录，以将配置数据导入 View 配置存储库。

- 2 在命令提示符处，键入 `vdmimport` 命令，并使用 `-f` 选项来指定要导入的 LDIF 配置文件。

例如：`vdmimport -f myexport.LDF`

在 `vdmimport` 命令运行后，将使用文件中的数据更新 View 连接服务器实例的配置，并显示已成功更新的记录数。如果因为您的角色没有足够的特权而无法更新某些记录，则会显示错误。

## 使用 LDIFDE 命令导入 LDAP 配置数据

您可以使用 Microsoft LDIFDE 命令将配置数据从 LDIF 配置文件导入标准或副本 View 连接服务器实例。

在 View 4.5 之前的 View 版本中，必须使用 Microsoft LDIFDE 命令从 LDIF 配置文件中导入配置数据。

View 4.5 之前的 View 版本不支持 `vdmimport` 命令。

如果使用的是 View 4.5 或更高版本，请使用 `vdmimport` 命令，而不是 LDIFDE 命令。`vdmimport` 命令不显示运行 LDIFDE 命令所生成的大量错误消息。有关更多信息，请参阅[使用 vdmimport 命令导入 LDAP 配置数据](#)。

由于 LDIFDE 命令不会更新、创建或删除任何未在 LDIF 配置文件中定义的 LDAP 记录，因此允许您自定义 LDIF 配置文件，以便在导入文件时只有选定的记录会受到影响。有关使用 LDIFDE 命令的完整信息，请转到 <http://support.microsoft.com/kb/237677>。

### 前提条件

将 LDAP 配置数据导出到 LDIF 配置文件。请参阅[导出 LDAP 配置数据](#)。

### 步骤

- 1 登录到 View 连接服务器实例。

选项	操作
View 3.1 或更低版本	以管理员身份登录并成为本地管理员用户组的成员。
View 4.5 或更高版本	以角色为管理员的用户身份登录。
	<b>注</b> 您必须以角色为管理员的用户身份登录，以将配置数据导入 View 配置存储库。

- 2 在命令提示符处，键入 LDIFDE 命令，并使用 `-f` 选项来指定现有的 LDIF 配置文件。

例如：LDIFDE `-i -f myexport.LDF -s 127.0.0.1 -z`

在 LDIFDE 命令运行后，将使用文件中的数据更新 View 连接服务器实例的配置，并显示已成功更新的记录数。

每当存储库中的现有条目被覆盖时，将会显示错误消息。可忽略这些错误消息。如果因为您的角色没有足够的特权而无法更新记录，也会显示错误消息。

# 将 View 与 Microsoft SCOM 相集成

# 5

您可以使用 Microsoft System Center Operations Manager (SCOM) 监控 View 组件的状态，包括 View 连接服务器实例、安全服务器，以及在 View 连接服务器主机和安全服务器主机上运行的 View 服务。

本章讨论了以下主题：

- 设置 SCOM 集成
- 在 Operations Manager 控制台中监控 View

## 设置 SCOM 集成

将 View 与 SCOM 集成的过程包括：为 View 连接服务器组分配名称、在 SCOM 服务器上导入 View 管理包、在每个 View 连接服务器实例和安全服务器上启用代理程序，以及在 Operations Manager 控制台中运行 View 发现脚本。

### 为 View 连接服务器组分配名称

您必须先在 View 中为 View 连接服务器组分配一个名称，然后才能使用 SCOM 监控和管理 View 组件的状态。Operations Manager 控制台会显示该名称，可帮助您在 SCOM 中识别 View 连接服务器组。

---

**注** View Administrator 不会显示 View 连接服务器组名称。

---

#### 前提条件

熟悉 vdmadmin 命令行界面。有关更多信息，请参阅《View 管理指南》文档。

#### 步骤

- 1 登录到 View 连接服务器组中的一个 View 连接服务器主机。
- 2 在命令提示符中，键入 vdmadmin 命令及 -C 和 -c 选项。

例如 `vdmadmin -C -c group_name`

-c 选项可指定要分配给 View 连接服务器组的名称。

#### 示例：分配 View 连接服务器组名称

在此示例中，vdmadmin 命令将 View 连接服务器组的名称设置为 VCSG01。

```
vdmadmin -C -c VCSG01
```

## 后续步骤

完成在 [SCOM 服务器上导入 View 管理包](#) 中描述的过程。

## View 管理包

通过 View 管理包，您可以使用 SCOM 监控和管理 View 组件的状态。

**表 5-1. View 管理包**

View 管理包	说明
VMware.View.Discovery.mp	包含发现 View Server 安装实例的代理。
VMware.View.Monitoring.mp	包含可在 Operations Manager 控制台中用于 View 的视图和监视器。
VMware.View.Library.mp	包含 View 中受管对象的类和关系定义。
VMware.View.Image.Library.mp	包含表示在 VMware.View.Library.mp 中定义的类的图形。

在安装 View 连接服务器软件时，View 管理包随之安装在 View 连接服务器实例或安全服务器上的 C:\Program Files\VMware\VMware View\Server\extras\ManagementPacks 目录中。

View 管理包需要随 SCOM 一起安装的默认系统管理包，以及 Microsoft Windows Server 基础操作系统版 System Center Operations Manager 2007 的管理包。

## 在 SCOM 服务器上导入 View 管理包

要使用 SCOM 监控和管理 View 组件的状态，您必须在 SCOM 服务器上导入 View 管理包。

**重要事项** McAfee VirusScan Enterprise 8.0i 会阻止 SCOM 使用的 Visual Basic 脚本运行。有关可用修补程序的更多详细信息，请访问 <http://support.microsoft.com/kb/890736/zh-cn>。

### 前提条件

- 完成 [为 View 连接服务器组分配名称](#) 中描述的过程。
- 熟悉 View 管理包。请参阅 [View 管理包](#)。

### 步骤

- 1 将 View 管理包从 View 连接服务器实例或安全服务器复制到 SCOM 服务器。  
View 管理包位于 View 连接服务器主机或安全服务器上的 C:\Program Files\VMware\VMware View\Server\extras\ManagementPacks 目录下。
- 2 在 Operations Manager 控制台中，转到 **管理\管理包**，然后选择 **导入管理包**。
- 3 使用“导入管理包”向导导入 View 管理包。

### 后续步骤

完成在 [View 连接服务器主机或安全服务器上启用代理程序](#) 中描述的过程。

## 在 View 连接服务器主机或安全服务器上启用代理程序

您必须使用 Operations Manager 控制台在您希望使用 SCOM 监视的每个 View 连接服务器主机或安全服务器上启用代理程序。只有在为服务器启用代理程序时，发现脚本才能发现 Windows Server。

### 前提条件

完成在 [SCOM 服务器上导入 View 管理包](#) 中描述的过程。

### 步骤

- 1 在 Operations Manager 控制台中，转到**管理\代理管理**，选择服务器，然后单击**属性**。
- 2 在**安全**选项卡中，选择**允许此代理充当代理并发现其他计算机上的托管对象**选项。
- 3 单击**确定**保存更改。

## 在 Operations Manager 控制台中运行发现脚本

发现脚本可查找安装有 View Server 的系统。它在 Windows Server 的注册表中探查指示 View 软件版本、服务器类型及 View 连接服务器组名称和 ID 的注册表项。

---

**注** 可选择手动运行发现脚本。发现脚本计划每小时自动运行一次。

---

### 前提条件

- 完成在 [SCOM 服务器上导入 View 管理包](#) 中描述的过程。
- 完成在 [View 连接服务器主机或安全服务器上启用代理程序](#)中描述的过程。只有在您使用 Operations Manager 控制台为服务器启用代理程序时，发现脚本才能发现 Windows Server。

### 步骤

- 1 在 Operations Manager 控制台中，转到**监视\Windows 计算机**。
- 2 选择计算机系统，然后单击 **VMware View 运行发现探查**操作。

如果发现脚本检测到计算机上安装有 View Server，它会创建在 `VMware.View.Library` 管理包中定义的 View 对象类实例，并在这些受管对象之间建立关系。

有关 View 连接服务器实例和安全服务器的受管对象列表，请参阅 [View 连接服务器和安全服务器受管对象](#)。有关 View 对象类及其关系的信息，请参阅 [View 对象类和关系](#)。

### 后续步骤

（可选）通过在 Operations Manager 控制台中查看对象，验证发现脚本为服务器创建的对象。请参阅[显示发现的受管 View 对象](#)。

## View 连接服务器和安全服务器受管对象

发现脚本可发现 View 连接服务器实例和安全服务器的受管对象。

**表 5-2. View 连接服务器实例和安全服务器的受管对象**

对象	View 连接服务器实例	安全服务器
VMware.View.Cluster	X	X
VMware.View.Cluster.Node.Item	X	X
VMware.View.ConnectionServerRole.Item	X	X
VMware.View.Component.ConnectionServer.Item	X	X
VMware.View.Component.Framework.Item	X	X
VMware.View.Component.Web.Item	X	
VMware.View.Component.Directory.Item	X	
VMware.View.Component.SecureGateway.Item	X	X
VMware.View.Component.MessageBus.Item	X	
VMware.View.Component.SecurityServer.Item		X

## View 对象类和关系

VMware.View.Library 管理包包含 View 管理包的类和关系定义。类可以包含属性，例如名称或 ID。类之间的关系描述其层次结构。例如，VMware.View.Clusters 和 VMware.View.Cluster 之间以及 VMware.View.Cluster 和 VMware.View.Cluster.Node 之间存在 contains 关系。

VMware.View.Library 管理包还包含类和属性的友好名称字符串。SCOM 控制台优先于类和属性名称显示友好名称。

## View 连接服务器组类

VMware.View.Library 管理包包含 View 连接服务器组类定义。

**表 5-3. View 库 View 连接服务器组类**

类名称	说明
VMware.View.Cluster	表示 View 连接服务器组。此类具有属性 ClusterID 和 DisplayName（组的名称）。
VMware.View.Clusters	表示包含 VMware.View.Cluster 实例的单独类。

## 基类

VMware.View.Library 管理包包含抽象基类定义。

**注** 当前支持的派生自这些类的实例必须为 View 5.1.x 或更高版本。



表 5-4. View 库基类

类名称	说明
<code>VMware.View.Cluster.Node</code>	表示 View 连接服务器组的成员。此类具有属性 <code>ClusterID</code> 、 <code>ClusterName</code> 、 <code>ProductVersion</code> 和 <code>InstallPath</code> 。
<code>VMware.View.Component</code>	表示已在 View 连接服务器组成员上安装的 View 组件。此类具有属性 <code>Name</code> 。
<code>VMware.View.Component.ConnectionServer</code>	表示已在 View 连接服务器组成员上安装的连接服务器组件。此类的属性继承自 <code>VMware.View.Component</code> 。
<code>VMware.View.Component.Directory</code>	表示已在 View 连接服务器组成员上安装的 Directory 组件。此类的属性继承自 <code>VMware.View.Component</code> 。
<code>VMware.View.Component.Framework</code>	表示已在 View 连接服务器组成员上安装的 Framework 组件。此类的属性继承自 <code>VMware.View.Component</code> 。
<code>VMware.View.Component.MessageBus</code>	表示已在 View 连接服务器组成员上安装的消息总线组件。此类的属性继承自 <code>VMware.View.Component</code> 。
<code>VMware.View.Component.SecurityGateway</code>	表示已在 View 连接服务器组成员上安装的安全网关组件。此类的属性继承自 <code>VMware.View.Component</code> 。
<code>VMware.View.Component.SecurityServer</code>	表示已在 View 连接服务器组成员上安装的安全服务器组件。此类的属性继承自 <code>VMware.View.Component</code> 。
<code>VMware.View.Component.Web</code>	表示已在 View 连接服务器组成员上安装的 Web 组件。此类的属性继承自 <code>VMware.View.Component</code> 。
<code>VMware.View.ConnectionServerRole</code>	表示装有连接服务器的 View 连接服务器组成员。此类的属性继承自 <code>VMware.View.NodeRole</code> 。
<code>VMware.View.NodeRole</code>	表示 View 连接服务器组成员的角色。
<code>VMware.View.SecurityServerRole</code>	表示装有安全服务器的 View 连接服务器组成员。此类的属性继承自 <code>VMware.View.NodeRole</code> 。

## 具体类

`VMware.View.Library` 管理包包含具体类定义。

**注** 这些具体类是最新版本，并受 View 5.1.x 及更高版本支持。

表 5-5. View 库的具体类

类名称	说明
<code>VMware.View.Cluster.Node.Item</code>	表示装有 View 5.1.x 版或更高版本的 View 连接服务器组成员。此类的属性继承自 <code>VMware.View.Cluster.Node</code> 。
<code>VMware.View.Component.ConnectionServer.Item</code>	表示已在 View 连接服务器组成员上安装的连接服务器组件 5.1.x 版或更高版本。此类的属性继承自 <code>VMware.View.Component.ConnectionServer</code> 。
<code>VMware.View.Component.Directory.Item</code>	表示已在 View 连接服务器组成员上安装的 Directory 组件 5.1.x 版或更高版本。此类的属性继承自 <code>VMware.View.Component.Directory</code> 。
<code>VMware.View.Component.Framework.Item</code>	表示已在 View 连接服务器组成员上安装的 Framework 组件 5.1.x 版或更高版本。此类的属性继承自 <code>VMware.View.Component.Framework</code> 。

类名称	说明
<code>VMware.View.Component.MessageBus.Item</code>	表示已在 View 连接服务器组成员上安装的 Message Bus 组件 5.1.x 版或更高版本。此类的属性继承自 <code>VMware.View.Component.MessageBus</code> 。
<code>VMware.View.Component.SecurityGateway.Item</code>	已在 View 连接服务器组成员上安装的 Gateway 组件。此类的属性继承自 <code>VMware.View.Component.SecurityGateway</code> 。
<code>VMware.View.Component.SecurityServer.Item</code>	表示已在 View 连接服务器组成员上安装的安全服务器组件 5.1.x 版或更高版本。此类的属性继承自 <code>VMware.View.Component.SecurityServer</code> 。
<code>VMware.View.Component.Web.Item</code>	表示已在 View 连接服务器组成员上安装的 Web 组件 5.1.x 版或更高版本。此类的属性继承自 <code>VMware.View.Component.Web</code> 。
<code>VMware.View.ConnectionServerRole.Item</code>	表示装有连接服务器 5.1.x 版或更高版本的 View 连接服务器组成员。此类的属性继承自 <code>VMware.View.NodeRole</code> 。
<code>VMware.View.SecurityServerRole.Item</code>	表示装有安全服务器 5.1.x 版或更高版本的 View 连接服务器组成员。此类的属性继承自 <code>VMware.View.NodeRole</code> 。

## 在 Operations Manager 控制台中监控 View

将 View 与 SCOM 相集成后，您可以使用 Operations Manager 控制台监控和管理 View 组件。

### 用于 View 的视图和监视器

`VMware.View.Monitoring` 管理包包含可用于在 Operations Manager 控制台中监控和管理 View 组件的视图和监视器。

#### 可用于 View 对象的视图

您可以使用 `VMware.View.Monitoring` 管理包中定义的视图查看发现的 View 对象。

**表 5-6. View 对象的可用视图**

View	说明
活动警示	显示严重的 View 警示。
节点状态	显示所有 View 连接服务器组的所有已发现成员的状态。
组状态	显示已发现的 View 连接服务器组的状态。
组	显示所有已发现的 View 连接服务器组、成员、角色和组件的图表。单击图标和连接器，可获取有关对象及其关系的详细信息。
连接服务器角色性能数据	显示以下数据集。 <ul style="list-style-type: none"> <li>■ 所有会话</li> <li>■ 所有高优先级会话</li> <li>■ SVI 会话</li> <li>■ 高优先级 SVI 会话</li> </ul>
安全网关角色性能数据	显示以下数据集。 <ul style="list-style-type: none"> <li>■ 安全网关会话</li> <li>■ 高优先级安全网关会话</li> </ul>

## View 对象的可用监视器类型

VMware.View.Monitoring 管理包提供以下监视器类型。

性能监视器	收集系统数据并将该数据返回到 SCOM 性能数据库和数据仓库。您可以在连接服务器角色性能数据和安全网关角色性能数据视图中以图形方式查看数据。
服务组件监视器	收集有关 View 组件服务状态的信息。如果受监控的服务未运行，SCOM 则将其状态设置为错误，并引发警示。如果某个组件处于错误状态，则受影响的 View 连接服务器组及其成员也会进入错误状态。
域连接监视器	确认 View 连接服务器实例可以绑定到该实例所属的所有域。此监视器每三分钟查询一次 View 连接服务器实例上 Web 组件的状态。如果 View 连接服务器实例无法绑定到域，SCOM 将其状态设置为错误，并引发警示。
事件数据库连接监视器	确认已配置事件数据库并且事件可写入到数据库中。此监视器每三分钟在 Web 组件中查询一次该信息，并且在事件数据库没有连接时引发警示。
Virtual Center (vCenter) 连接监视器	确认 View 连接服务器实例可以连接到配置的 vCenter Server 实例。此监视器每三分钟在 Web 组件中查询一次该信息，并且在 vCenter Server 实例不可用时引发警示。

## View 连接服务器实例的服务组件监视器

下表介绍了 VMware.View.Monitoring 管理包为 View 连接服务器实例提供的服务组件监视器。

表 5-7. View 连接服务器实例的 View 服务组件监视器

监视器	显示名称	受监控服务
ConnectionServerServiceCheck	连接服务器服务运行状况	VMware View 连接服务器
FrameworkServiceCheck	基础框架服务运行状况	VMware View Framework 组件
MessageBusServiceCheck	消息总线服务运行状况	VMware View Message Bus 组件
SecureGatewayCheck	安全网关服务运行状况	VMware View Security Gateway 组件
WebServiceCheck	Web 服务运行状况	VMware View Web 组件
DirectoryServiceCheck	目录服务运行状况	VMwareVDMDS

## 安全服务器的服务组件监视器

下表介绍了 VMware.View.Monitoring 管理包为安全服务器提供的服务组件监视器。

表 5-8. 安全服务器的 View Server 组件监视器

监视器	显示名称	受监控服务
SecureGatewayServerServiceCheck	安全服务器服务运行状况	VMware View 安全服务器
FrameworkServiceCheck	基础框架服务运行状况	VMware View Framework 组件
SecureGatewayCheck	安全网关服务运行状况	VMware View Security Gateway 组件

## 显示发现的受管 View 对象

您可以在 Operations Manager 控制台中显示发现的受管 View 对象。

### 前提条件

将 View 与 SCOM 相集成。请参阅[设置 SCOM 集成](#)。

### 步骤

- ◆ 要显示发现脚本为服务器创建的 View 对象，请在 Operations Manager 控制台中转到[监视\发现清单](#)。
- ◆ 要显示 SCOM 管理的 View 对象以及这些对象之间的关系，请在 Operations Manager 控制台中转到[监视\VMware View](#)，然后选择所需的视图。

## 显示性能信息

您可以在 Operations Manager 控制台中显示 View 连接服务器实例或安全服务器的图形性能数据。

### 前提条件

将 View 与 SCOM 相集成。请参阅[设置 SCOM 集成](#)。

### 步骤

- 1 在 Operations Manager 控制台中，转到[监视\VMware View\性能](#)。
- 2 选择[连接服务器角色的性能数据](#)或[安全网关角色的性能数据](#)视图。
- 3 选择所需的数据集。

## 显示 View 连接服务器组的警示

您可以使用 Operations Manager 控制台中的运行状况资源管理器，来显示运行状况监视器针对 View 连接服务器组发出的警示的信息。

### 前提条件

将 View 与 SCOM 相集成。请参阅[设置 SCOM 集成](#)。

### 步骤

- 1 在 Operations Manager 控制台中，转到[监视\VMware View](#)，然后选择[活动警示](#)视图。
- 2 选择警示可显示有关该警示的知识文章。

## 关闭警示

对于 Operations Manager 控制台中的警示，您可以将其关闭，而不执行任何操作。

---

**注** 如果根本原因仍然存在，那么即使采用此方法，也会再次出现警示。

---

### 前提条件

将 View 与 SCOM 相集成。请参阅[设置 SCOM 集成](#)。

**步骤**

- 1 在 Operations Manager 控制台中，转到**监视\VMware View**，然后选择**活动警示**视图。
- 2 选择相应警示，然后单击**关闭警示**操作。

**重新启动 View 组件服务**

您可以从 Operations Manager 控制台的运行状况资源管理器中重新启动服务。如果 View 组件服务停止工作，服务组件监视器将发出警示。

**前提条件**

将 View 与 SCOM 相集成。请参阅[设置 SCOM 集成](#)。

**步骤**

- 1 在 Operations Manager 控制台中，转到**监视\VMware View**，然后选择**组状态**视图或**组节点状态**视图。
- 2 右键单击处于警示状态的 View 连接服务器组或成员，并选择**打开 > 运行状况资源管理器**。
- 3 在运行状况资源管理器中，选择警示，然后单击知识文章中的**重新启动服务**。

**从连接监控中排除域**

域连接运行状况监视器检查 View 连接服务器主机所在域与任何受信任域之间的连接。如果不希望看到有关某个域的警示，您可以从连接监控中排除该域。

**前提条件**

将 View 与 SCOM 相集成。请参阅[设置 SCOM 集成](#)。

**步骤**

- 1 在 Operations Manager 控制台中，转到**监视\VMware View**，然后选择**组状态**视图或**组节点状态**视图。
- 2 右键单击 View 连接服务器实例，并选择**打开 > 运行状况资源管理器**。
- 3 在运行状况资源管理器中右键单击 View 连接服务器实例的**域连接运行状况**条目，然后选择**监视器属性**。
- 4 在**替代**选项卡中，单击**替代**，然后为同一类的所有对象选择该选项。
- 5 在**替代属性**窗口中，选中 DomainExcludeList 参数对应的**替代**复选框，在**替代设置**文本框中键入排除域的名称，然后选中**强制**复选框。  
要排除多个域，请用空格分隔域名。
- 6 单击**应用**，然后单击**确定**以保存更改。

# 使用 WMI 查看 PCoIP 会话统计信息

# 6

您可以通过任何一种受支持的编程接口（包括 C#、C++、PowerShell、VBScript、VB .NET）和 Windows Management Instrumentation 命令行 (Management Instrumentation Command-line, WMIC)，使用 Windows Management Instrumentation (WMI) 来查看 PCoIP 会话的性能统计信息。

您还可以使用 Microsoft WMI Code Creator 工具生成可访问 PCoIP 性能计数器的 VBScript、C# 和 VB .NET 代码。有关 WMI、WMIC 和 WMI Code Creator 工具的更多信息，请访问 <http://technet.microsoft.com/zh-cn/library/bb742610.aspx> 和 <http://www.microsoft.com/downloads/cn/details.aspx?familyid=2cc30a64-ea15-4661-8da4-55bbc145c30e&displaylang=cn>。

本章讨论了以下主题：

- 使用 PCoIP 会话统计信息
- 常规 PCoIP 会话统计信息
- PCoIP 音频统计信息
- PCoIP 图像处理统计信息
- PCoIP 网络统计信息
- PCoIP USB 统计信息
- 使用 PowerShell cmdlet 查看 PCoIP 统计信息的示例

## 使用 PCoIP 会话统计信息

PCoIP 会话统计信息的 WMI 命名空间为 `root\CIMV2`。根据统计信息是在 PCoIP Server 还是 PCoIP 客户端上记录的，统计信息的名称采用 (Server) 或 (Client) 作为后缀。

您可以使用 Windows 性能监控器 (PerfMon) 通过计数器来计算指定采样周期内的平均值。您必须具有管理员特权才能远程访问性能计数器。

PCoIP 会话关闭后，所有统计信息将重置为 0。如果 WMI `SessionDurationSeconds` 属性是一个非零值并保持不变，则表示 PCoIP Server 已被强制结束或崩溃。如果 `SessionDurationSeconds` 属性从一个非零值更改为 0，则表示 PCoIP 会话已关闭。

为避免除数为零错误，请确认用于计算带宽或数据包丢失百分比的表达式中的分母不为零。

将记录零客户端的 USB 统计信息，但不记录瘦客户端或软件客户端的 USB 统计信息。

## 常规 PCoIP 会话统计信息

常规 PCoIP 会话统计信息的 WMI 类名称为 Win32\_PerfRawData\_TeradiciPerf\_PCoIPSessionGeneralStatistics。

**表 6-1. 常规会话统计信息**

WMI 属性名称	说明
BytesReceived	自 PCoIP 会话启动以来接收的 PCoIP 数据的总字节数。
BytesSent	自 PCoIP 会话启动以来传输的 PCoIP 数据的总字节数。
PacketsReceived	自 PCoIP 会话启动以来成功接收的数据包总数。并非所有数据包均具有相同的大小。
PacketsSent	自 PCoIP 会话启动以来传输的数据包总数。并非所有数据包均具有相同的大小。
RXPacketsLost	自 PCoIP 会话启动以来丢失的已接收数据包总数。
SessionDurationSeconds	PCoIP 会话处于打开状态的总秒数。
TXPacketsLost	自 PCoIP 会话启动以来丢失的已传输数据包总数。

### 计算用于接收的 PCoIP 数据的带宽

要计算时间 t1 到时间 t2 的时间间隔内用于接收的 PCoIP 数据的带宽（以千字节/秒为单位），请使用以下公式。

$$(\text{BytesReceived}[t2] - \text{BytesReceived}[t1]) * 8 / (1024 * (t2 - t1))$$

### 计算用于传输的 PCoIP 数据的带宽

要计算时间 t1 到时间 t2 的时间间隔内用于传输的 PCoIP 数据的带宽（以千比特/秒为单位），请使用以下公式。

$$(\text{BytesSent}[t2] - \text{BytesSent}[t1]) * 8 / (1024 * (t2 - t1))$$

### 计算接收的 PCoIP 数据的数据包丢失

要计算已接收数据包的数据包丢失比例，请使用以下公式。

$$100 / (1 + ((\text{PacketsReceived}[t2] - \text{PacketsReceived}[t1]) / (\text{RXPacketsLost}[t2] - \text{RXPacketsLost}[t1])))$$

### 计算传输的 PCoIP 数据的数据包丢失

要计算已传输数据包的数据包丢失比例，请使用以下公式。

$$100 * (\text{TXPacketsLost}[t2] - \text{TXPacketsLost}[t1]) / (\text{PacketsSent}[t2] - \text{PacketsSent}[t1])$$

## PCoIP 音频统计信息

PCoIP 音频统计信息的 WMI 类名称为

Win32\_PerfRawData\_TeradiciPerf\_PCoIPSessionAudioStatistics。

**注** 音频统计信息不包括 USB 数据内携带的音频数据。

**表 6-2. PCoIP 音频统计信息**

WMI 属性名称	说明
AudioBytesReceived	自 PCoIP 会话启动以来接收的音频数据的总字节数。
AudioBytesSent	自 PCoIP 会话启动以来发送的音频数据的总字节数。
AudioRXBkbitPersec	采样周期（以秒为单位）内用于入站音频数据包的平均带宽。
AudioTXBkbitPersec	采样周期（以秒为单位）内用于出站音频数据包的平均带宽。
AudioTXBWLimitkbitPersec	出站音频数据包的传输带宽限制，以千比特/秒为单位。此限制由 GPO 设置定义。

### 计算用于接收的音频数据的带宽

要计算时间 t1 到时间 t2 的时间间隔内用于接收的音频数据的带宽（以千比特/秒为单位），请使用以下公式。

$$(\text{AudioBytesReceived}[t2] - \text{AudioBytesReceived}[t1]) * 8 / (1024 * (t2 - t1))$$

此计算不使用 AudioTXBkbitPersec。

### 计算用于传输的音频数据的带宽

要计算时间 t1 到时间 t2 的时间间隔内用于传输的音频数据的带宽（以千比特/秒为单位），请使用以下公式。

$$(\text{AudioBytesSent}[t2] - \text{AudioBytesSent}[t1]) * 8 / (1024 * (t2 - t1))$$

此计算不使用 AudioTXBkbitPersec。

## PCoIP 图像处理统计信息

PCoIP 图像处理统计信息的 WMI 类名称为

Win32\_PerfRawData\_TeradiciPerf\_PCoIPSessionImagingStatistics。

**表 6-3. PCoIP 图像处理统计信息**

WMI 属性名称	说明
ImagingBytesReceived	自 PCoIP 会话启动以来接收的图像处理数据的总字节数。
ImagingBytesSent	自 PCoIP 会话启动以来传输的图像处理数据的总字节数。



WMI 属性名称	说明
ImagingDecoderCapabilitykbitPersec	图像理解码器的估计处理能力，以千比特/秒为单位。此统计信息每秒钟更新一次。
ImagingEncodedFramesPersec	在为时一秒的采样周期内编码的图像处理帧数。
ImagingActiveMinimumQuality	以 0 至 100 范围为尺度的最低编码质量值。此统计信息每秒钟更新一次。此计数器不对应于最低质量的 GPO 设置。
ImagingRXBkbitPersec	采样周期（以秒为单位）内用于入站图像处理数据包的平均带宽。
ImagingTXBkbitPersec	采样周期（以秒为单位）内用于出站图像处理数据包的平均带宽。

## 计算用于接收的图像处理数据的带宽

要计算时间 **t1** 到时间 **t2** 的时间间隔内用于接收的图像处理数据的带宽（以千比特/秒为单位），请使用以下公式。

$$(\text{ImagingBytesReceived}[t2] - \text{ImagingBytesReceived}[t1]) * 8 / (1024 * (t2 - t1))$$

此计算不使用 ImagingRXBkbitPersec。

## 计算用于传输的图像处理数据的带宽

要计算时间 **t1** 到时间 **t2** 的时间间隔内用于传输的图像处理数据的带宽（以千比特/秒为单位），请使用以下公式。

$$(\text{ImagingBytesSent}[t2] - \text{ImagingBytesSent}[t1]) * 8 / (1024 * (t2 - t1))$$

此计算不使用 ImagingTXBkbitPersec。

## PCoIP 网络统计信息

PCoIP 网络统计信息的 WMI 类名称为

Win32\_PerfRawData\_TeradiciPerf\_PCoIPSessionNetworkStatistics。

**表 6-4. PCoIP 网络统计信息**

WMI 属性名称	说明
RoundTripLatencymsec	PCoIP Server 和 PCoIP 客户端之间的往返延迟，以毫秒为单位。
RXBkbitPersec	采样周期（以秒为单位）内用于入站 PCoIP 数据包的总平均带宽。
RXBWPeakkbitPersec	在为期一秒的采样周期内入站 PCoIP 数据包的峰值带宽，以千比特/秒为单位。
RXPacketLossPercent	采样周期内接收的数据包的丢失百分比。
TXBkbitPersec	采样周期（以秒为单位）内用于出站 PCoIP 数据包的总平均带宽。

WMI 属性名称	说明
TXBWActiveLimitkbitPersec	估计的可用网络带宽，以千比特/秒为单位。此统计信息每秒钟更新一次。
TXBWLimitkbitPersec	出站数据包的传输带宽限制，以千比特/秒为单位。此限制是以下值的最小值。 <ul style="list-style-type: none"> <li>■ PCoIP 客户端的 GPO 带宽限制</li> <li>■ PCoIP Server 的 GPO 带宽限制</li> <li>■ 本地网络连接的带宽限制</li> <li>■ 基于加密限制的零客户端固件协商带宽限制</li> </ul>
TXPacketLossPercent	采样周期内传输的数据包的丢失百分比。

## 计算用于接收的网络数据的带宽

要计算时间 **t1** 到时间 **t2** 的时间间隔内用于接收的数据的带宽（以千比特/秒为单位），请使用以下公式。

$$(\text{BytesReceived}[t2] - \text{BytesReceived}[t1]) * 8 / (1024 * (t2 - t1))$$

此计算不使用 RXBWkbitPersec。

## 计算用于传输的网络数据的带宽

要计算时间 **t1** 到时间 **t2** 的时间间隔内用于传输的数据的带宽（以千比特/秒为单位），请使用以下公式。

$$(\text{BytesSent}[t2] - \text{BytesSent}[t1]) * 8 / (1024 * (t2 - t1))$$

此计算不使用 TXBWkbitPersec。

## 计算接收的网络数据的数据包丢失

要计算时间 **t1** 到时间 **t2** 的时间间隔内接收的数据的数据包丢失百分比，请使用以下公式。

$$\text{PacketsReceived during interval} = (\text{PacketsReceived}[t2] - \text{PacketsReceived}[t1])$$

$$\text{RXPacketsLost during interval} = (\text{RXPacketsLost}[t2] - \text{RXPacketsLost}[t1])$$

$$\text{RXPacketsLost \%} = \text{RXPacketsLost during interval} / (\text{RXPacketsLost during interval} + \text{PacketsReceived during interval}) * 100$$

此计算不使用 RXPacketLostPercent 或 RXPacketLostPercent\_Base。

## 计算传输的网络数据的数据包丢失

要计算时间 **t1** 到时间 **t2** 的时间间隔内传输的数据的数据包丢失百分比，请使用以下公式。

$$\text{PacketsSent during interval} = (\text{PacketsSent}[t2] - \text{PacketsSent}[t1])$$

$$\text{TXPacketsLost during interval} = (\text{TXPacketsLost}[t2] - \text{TXPacketsLost}[t1])$$

```
TXPacketsLost % = TXPacketsLost during interval /
(TXPacketsLost during interval + PacketsSent during interval) * 100
```

此计算不使用 TXPacketLostPercent 或 TXPacketLostPercent\_Base。

使用此公式可防止数据包丢失百分比高于 100%。由于 PacketsLost 和 PacketsSent 以异步方式获取，因此需要该计算。

## PCoIP USB 统计信息

PCoIP USB 统计信息的 WMI 类名称为

Win32\_PerfRawData\_TeradiciPerf\_PCoIPSessionUSBStatistics。

**表 6-5. PCoIP USB 统计信息**

WMI 属性名称	说明
USBBytesReceived	自 PCoIP 会话启动以来接收的 USB 数据的总字节数。
USBBytesSent	自 PCoIP 会话启动以来传输的 USB 数据的总字节数。
USBRXBkbitPersec	采样周期（以秒为单位）内用于入站 USB 数据包的平均带宽。
USBTXBkbitPersec	采样周期（以秒为单位）内用于出站 USB 数据包的平均带宽。

### 计算用于接收的 USB 数据的带宽

要计算时间 t1 到时间 t2 的时间间隔内用于接收的 USB 数据的带宽（以千比特/秒为单位），请使用以下公式。

```
(USBBytesReceived[t2]-USBBytesReceived[t1]) * 8 / (1024 * (t2-t1))
```

此计算不使用 USBRXBkbitPersec。

### 计算用于传输的 USB 数据的带宽

要计算时间 t1 到时间 t2 的时间间隔内用于传输的 USB 数据的带宽（以千比特/秒为单位），请使用以下公式。

```
(USBBytesSent[t2]-USBBytesSent[t1]) * 8 / (1024 * (t2-t1))
```

此计算不使用 USBTXBkbitPersec。

## 使用 PowerShell cmdlet 查看 PCoIP 统计信息的示例

您可以使用 PowerShell cmdlet 查看 PCoIP 统计信息。

在以下示例中，Get-WmiObject cmdlet 检索客户端 cm-02 的 PCoIP 网络统计信息。

```
Get-WmiObject -namespace "root\cimv2" -computername cm-02 -class
Win32_PerfRawData_TeradiciPerf_PCoIPSessionNetworkStatistics
```

在以下示例中，`Get-WmiObject cmdlet` 检索发生任何传输数据包丢失时桌面 `dt-03` 的 PCoIP 常规会话统计信息。

```
Get-WmiObject -namespace "root\cimv2" -computername desktop-03 -query "select * from Win32_PerfRawData_TeradiciPerf_PCoIPSessionGeneralStatistics where TXPacketsLost > 0"
```

## 使用启动会话脚本设置桌面策略

通过启动会话脚本，您可以根据从 **Horizon Client** 和 **View** 连接服务器收到的信息，在桌面会话启动之前配置特定的 **View** 桌面设置。

例如，您可以使用启动会话脚本配置基于客户端设备和用户位置的桌面策略，而不必设置多个具有不同桌面策略的桌面池。启动会话脚本可为 IP 地址属于组织内部域的用户启用映射的驱动器、剪贴板重定向和其他桌面功能，但对那些 IP 地址属于外部域的用户禁用这些功能。

本章讨论了以下主题：

- 获取启动会话脚本的输入数据
- 使用启动会话脚本的最佳做法
- 准备 **View** 桌面以使用启动会话脚本
- 示例启动会话脚本

### 获取启动会话脚本的输入数据

启动会话脚本无法以交互方式运行。启动会话脚本在 **View** 创建的环境中运行，并且必须从该环境中获取它的输入数据。

启动会话脚本从客户端计算机上的环境变量中收集输入数据。启动会话环境变量的前缀为 **VDM\_StartSession\_**。例如，包含客户端系统 IP 地址的启动会话环境变量为 **VDM\_StartSession\_IP\_Address**。您必须确保启动会话脚本验证它使用的任何环境变量存在与否。

有关类似于启动会话环境变量的变量列表，请参阅《在 **View** 中设置桌面和应用程序池》文档中的“发送到 **View** 桌面的客户端系统信息”。

### 使用启动会话脚本的最佳做法

使用启动会话脚本时，请采用以下最佳做法。

#### 何时使用启动会话脚本

只有当您需要在会话启动前先配置桌面策略时，才应使用启动会话脚本。

最佳做法是，在连接或重新连接桌面会话后，使用 **Horizon AgentCommandsToRunOnConnect** 和 **CommandsToRunOnReconnect** 组策略设置运行命令脚本。在大多数用例中，在桌面会话内运行脚本即可，而无需使用启动会话脚本。

有关更多信息，请参阅《在 View 中设置桌面和应用程序池》文档中的“在 View 桌面上运行命令”。

## 管理启动会话超时

确保您的启动会话脚本可快速运行。

如果您在 Windows 注册表中设置了 `WaitScriptsOnStartSession` 值，则必须先完成启动会话脚本运行，然后 Horizon Agent 才能对 View 连接服务器发出的 `StartSession` 消息做出响应。如果脚本的运行时间较长，可能会导致 `StartSession` 请求超时。

如果发生超时，并且池使用浮动分配，那么 View 连接服务器会尝试将用户连接到另一台虚拟机。如果发生超时，并且没有虚拟机可用，则 View 连接服务器将拒绝用户的连接请求。

最佳做法是，为脚本主机操作设置硬超时，以使系统在脚本运行时间过长时可以返回特定错误。

## 使启动会话脚本可供访问

您配置启动会话脚本的路径必须仅可由系统帐户和本地管理员访问。设置 ACL 以使基础密钥仅可由上述帐户访问。

最佳做法是，将启动会话脚本放在 `View_Agent_install_path\scripts` 目录中，例如：

```
%ProgramFiles%\VMware\VMware View\Agent\scripts\sample.vbs
```

默认情况下，此目录仅可由系统帐户和管理员帐户访问。

## 准备 View 桌面以使用启动会话脚本

要准备 View 桌面以使用启动会话脚本，必须启用 VMware View 脚本主机服务，并在 Windows 注册表中添加相关项。

您必须配置所有需要运行启动会话脚本的 View 桌面。View 不提供将注册表更改、VMware View 脚本主机服务配置更改和启动会话脚本传播到多个 View 桌面虚拟机的机制。

## 启用 VMware View 脚本主机服务

在您希望 View 运行启动会话脚本的每个 View 桌面虚拟机上，您必须启用 VMware View 脚本主机服务。默认情况下，将禁用 VMware View 脚本主机服务。

配置 VMware View 脚本主机服务时，您可以选择指定启动会话脚本运行时所用的用户帐户。启动会话脚本在 VMware View 脚本主机服务的上下文中运行。默认情况下，VMware View 主机脚本服务会配置为以系统用户身份运行。

---

**重要事项** 启动会话脚本在桌面用户会话之外运行，且不由桌面用户帐户运行。在以系统用户身份运行的脚本中，会直接从客户端计算机发送信息。

---

### 步骤

- 1 登录到 View 桌面虚拟机。
- 2 在命令提示符中，键入 `services.msc` 以启动 Windows 服务工具。

- 3 在详细信息窗格中，右键单击 **VMware View** 脚本主机服务条目，然后选择**属性**。
- 4 在**常规**选项卡上，从**启动类型**下拉菜单中选择**自动**。
- 5 （可选） 如果不希望本地系统帐户运行启动会话脚本，请选择**登录**选项卡，选择**此帐户**，然后键入要运行启动会话脚本的帐户的用户名和密码。
- 6 单击**确定**，并退出 Windows 服务工具。

## 为启动会话脚本添加 Windows 注册表项

您必须在希望 View 运行启动会话脚本的每个 View 桌面虚拟机上添加 Windows 注册表项。

### 前提条件

- 确认您配置启动会话脚本的路径仅可供系统帐户和本地管理员访问。有关更多信息，请参阅[使启动会话脚本可供访问](#)。
- 确保您的启动会话脚本可快速运行。如果您在 Windows 注册表中设置了 **WaitScriptsOnStartSession** 值，则必须先完成启动会话脚本运行，然后 **Horizon Agent** 才能对 View 连接服务器发出的 **StartSession** 消息做出响应。有关更多信息，请参阅[管理启动会话超时](#)。

### 步骤

- 1 登录到 View 桌面虚拟机。
- 2 在命令提示符中，键入 **regedit** 以启动 Windows 注册表编辑器。
- 3 在注册表中，导航到 **HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents**。
- 4 将启动会话脚本的路径添加到注册表中。
  - a 在导航区域中，右键单击 **ScriptEvents**，选择 **新建 > 项**，然后创建一个名为 **StartSession** 的项。
  - b 在导航区域中，右键单击 **StartSession**，选择**新建 > 字符串值**，然后创建一个字符串值以标识要运行的启动会话脚本，例如，**SampleScript**。

要运行多个启动会话脚本，请在 **StartSession** 项下为每个脚本创建一个字符串值条目。您无法指定这些脚本的运行顺序。如果脚本必须按特定顺序运行，请从单个控制脚本调用它们。
  - c 在主题区域中，右键单击新字符串值条目，然后选择**修改**。
  - d 在**数值数据**文本框中，键入用于调用启动会话脚本的命令行，然后单击**确定**。

键入启动会话脚本及其所需的任何文件的完整路径。
- 5 在注册表中添加并启用启动会话值。
  - a 导航到 **HKLM\SOFTWARE\VMware, Inc.\VMware VDM\Agent\Configuration**。
  - b （可选） 如果 **Configuration** 项不存在，请右键单击 **Agent**，选择**新建 > 项**，然后创建该项。
  - c 在导航区域中，右键单击 **Configuration**，选择**新建 > DWORD (32 位) 值**，然后键入 **RunScriptsOnStartSession**。
  - d 在主题区域中，右键单击新 **DWORD** 值条目，然后选择**修改**。

- e 在**数值数据**文本框中，键入 1 以启用启动会话脚本，然后单击**确定**。

键入 0 可禁用此功能。默认值为 0。

- f （可选）要使 Horizon Agent 延迟发送 StartSession 响应，请向 Configuration 项再添加一个名为 WaitScriptsOnStartSession 的 DWORD 值。

如果 WaitScriptsOnStartSession 数据值为 1，则会使 Horizon Agent 延迟发送 StartSession 响应，并在脚本未完成时失败。值为 0 表示 Horizon Agent 不等待脚本完成或检查脚本退出代码就发送 StartSession 响应。默认值为 0。

- 6 设置一个注册表值以指定可防止脚本超时的超时值（以秒为单位，而不是分钟）。

通过设置此以秒为单位的超时值，可以用秒作为单位来配置 VMware View 脚本主机服务超时值。例如，如果将 VMware View 脚本主机服务的超时值设置为 30 秒，可以确保在 View 连接服务器发生超时之前，启动会话脚本或者完成运行，或者超时。

- a 导航到 HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents。

- b 添加名为 TimeoutsInMinutes 的 DWORD 值。

- c 将数据值设置为 0。

- 7 （可选）要启用 VMware View 脚本主机服务以使启动会话脚本超时，请设置超时值。

- a 导航到 HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents\StartSession。

- b 在主题区域中，右键单击 Default (@) 项，然后选择**修改**。

- c 在**数值数据**文本框中，键入超时值，然后单击**确定**。

值为 0 表示未设置超时。

- 8 退出注册表编辑器，并重新启动系统。

## 示例启动会话脚本

这些示例启动会话脚本描述了如何将环境变量写入文件，测试超时功能和测试非零退出代码。

以下示例 Visual Basic 脚本将提供给脚本的所有环境变量写入文件。您可以在您自己的环境中使用此示例脚本查看示例数据。您可以将此脚本保存为 C:\sample.vbs。

```
Option Explicit
Dim WshShell, FSO, outFile, strOutputFile, objUserEnv, strEnv

strOutputFile = "c:\setvars.txt"

Set FSO = CreateObject("Scripting.FileSystemObject")
Set outFile = FSO.CreateTextFile(strOutputFile, TRUE)
outFile.WriteLine("Script was called at (" & Now & ")")

Set WshShell = CreateObject( "WScript.Shell" )
Set objUserEnv = WshShell.Environment("PROCESS")
For Each strEnv In objUserEnv
    outFile.WriteLine(strEnv)
Next
```



```
outFile.Close
```

以下示例脚本测试超时功能。

```
Option Explicit  
WScript.Sleep 60000
```

以下示例脚本测试非零退出代码。

```
Option Explicit  
WScript.Quit 2
```