# View Integration

VMware Horizon 6.0

**vm**ware®

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

# Contents

# View Integration

This guide describes how to integrate View™ software with third-party software such as
Windows PowerShell, business intelligence reporting engines, and Microsoft System Center Operations
Manager (SCOM).

## Intended Audience

This book is intended for anyone who wants to customize or integrate software to work with View. The
information in this manual is written for experienced Windows or Linux system administrators who are
familiar with virtual machine technology and datacenter operations.

# Introduction

# 1

With View, system administrators can provision desktops and control user access to these desktops. Client software connects users to virtual machines running in VMware vSphere™, or to physical systems running within your network environment. In addition, View administrators can configure Remote Desktop Services (RDS) hosts to provide View desktop and application sessions to client devices.

This chapter includes the following topics:

■ "View Components" on page 9

■ "Integration Interfaces to View" on page 10

## View Components

You can use View with VMware vCenter Server to create desktops from virtual machines that are running on VMware ESX® or VMware ESXi™ hosts and deploy these desktops to end users. You can also install View on RDS hosts to deploy desktops and applications to end users. View uses your existing Active Directory infrastructure for user authentication and management.

After you create a desktop or application, authorized end users can use Web-based or locally installed client software to securely connect to centralized virtual machines, back-end physical systems, or Remote Desktop Services (RDS) hosts.

View consists of the following major components:

■ View Connection Server – a software service that acts as a broker for client connections by authenticating and then directing incoming user requests to the appropriate virtual machine, physical system, or Remote Desktop Services (RDS) host.

■ View Agent – a software service that is installed on all guest virtual machines, physical systems, or RDS hosts in order to allow them to be managed by View. The agent provides features such as connection monitoring, Virtual Printing, USB support, and single sign-on.

■ Horizon Client – a software application that communicates with View Connection Server to allow users to connect to their desktops.

■ View Administrator – a Web application that allows View administrators to configure View Connection Server, deploy desktop and application pools, manage machines, control user authentication, initiate and examine system events, and carry out analytical activities.

■ vCenter Server – a server that acts as a central administrator for ESX/ESXi hosts that are connected on a network. A vCenter Server provides the central point for configuring, provisioning, and managing virtual machines in the datacenter.

■ View Composer – a software service that is installed on a vCenter server to allow View to rapidly deploy multiple linked-clone desktops from a single centralized base image.

# Integration Interfaces to View

You can use several interfaces to integrate View with external applications.

- Event database – you can configure View to record events to a Microsoft SQL Server or Oracle database. You can then use business intelligence reporting engines to access and analyze this database.

- View PowerCLI – you can use the PowerShell interface to perform a wide variety of administration tasks on View components.

- Lightweight Directory Access Protocol (LDAP) – you can export and import LDAP configuration data from and into View. You can create scripts that update this configuration data without accessing View Administrator directly.

- Microsoft System Center Operations Manager (SCOM) – you can monitor the operations of View components from the SCOM console.

- Windows Management Instrumentation (WMI) – you can examine performance statistics for a PCoIP session.

# Integrating with the Event Database  2

You can configure View to record events to a Microsoft SQL Server or Oracle database. View records events such as the following examples:

- End-user actions such as logging in and starting a desktop or application session.
- Administrator actions such as adding entitlements and creating desktop or application pools.
- Alerts that report system failures and errors.
- Statistical sampling such as recording the maximum number of users over a 24-hour period.

You can use business intelligence reporting engines such as Crystal Reports, IBM Cognos, MicroStrategy 9, and Oracle Enterprise Performance Management System to access and analyze the event database.

This chapter includes the following topics:

- "Event Database Tables and Schemas" on page 11
- "Connection Broker Events" on page 13
- "View Agent Events" on page 19
- "View Administrator Events" on page 19
- "Event Message Attributes" on page 26
- "Sample Database Queries and Views" on page 28

## Event Database Tables and Schemas

Table 2-1 shows the tables that implement the event database in View.

**Table 2-1.** Event Database Tables

| Table Name | Description |
| --- | --- |
| event | Metadata and search optimization data for recent events. |
| event_data | Data values for recent events. |
| event_data_historical | Data values for all events. |
| event_historical | Metadata and search optimization data for all events. |

The event database prepends the names of these tables with a prefix that you define when you set up the database. See the *View Installation* document for more information.

**IMPORTANT**   View does not restrict the growth of the event_historical and event_data_historical tables. You must implement a space management policy for these tables.

View records details about events to all the database tables. After a certain period of time has elapsed since writing an event record, View deletes the record from the event and event_data tables. You can use View Administrator to configure the time period for which the database keeps a record in the event and event_data tables. See the *View Installation* document for more information.

---

NOTE   It is possible for events to be lost if you restart View Connection Server instances while the event database is not running. For a solution that avoids this problem see http://kb.vmware.com/kb/1021461.

---

A unique primary key, EventID, identifies each event that View records in the event and event_historical tables. View records data values for each event in the event_data and event_data_historical tables. You can obtain the complete set of information for an event by joining the event and event_data tables or the event_historical and event_data_historical tables on the EventID column.

The EventType, Severity, and Time columns in the event and event_historical tables identify the type and severity of an event and the time at which it occurred.

Table 2-2 shows the schema for the event and event_historical tables.

**Table 2-2.**  Schema for the event and event_historical Tables

| Column Name | Oracle Data Type | SQLServer Data Type | Description |
|---|---|---|---|
| Acknowledged | SMALLINT | tinyint | Whether View acknowledged the event. 0 = false 1 = true |
| DesktopId | NVARCHAR2(512) | nvarchar(512) | Desktop ID of the associated pool. |
| EventID | INTEGER | int | Unique primary key for the event. |
| EventType | NVARCHAR2(512) | nvarchar(512) | Event name that corresponds to an item in the message catalog. For example, BROKER_USERLOGGEDIN. |
| FolderPath | NVARCHAR2(512) | nvarchar(512) | Full path of the folder that contains the associated object. |
| GroupId | NVARCHAR2(512) | nvarchar(512) | SID of the associated group in Active Directory. |
| LUNId | NVARCHAR2(512) | nvarchar(512) | ID of the LUN that stores the associated object. |
| MachineId | NVARCHAR2(512) | nvarchar(512) | ID of the associated physical or virtual machine. |
| Module | NVARCHAR2(512) | nvarchar(512) | View component that raised the event. For example, Admin, Broker, Tunnel, Framework, Client, or Agent. |
| ModuleAndEventText | NVARCHAR2(512) | nvarchar(512) | Event message with values substituted for attribute parameters. |
| Node | NVARCHAR2(512) | nvarchar(512) | Name of the virtual device node. |
| Severity | NVARCHAR2(512) | nvarchar(512) | Severity level. For example, INFO, WARNING, ERROR, AUDIT_SUCCESS, AUDIT_FAIL. |
| Source | NVARCHAR2(512) | nvarchar(512) | Identifier for the source of the event. |
| ThinAppId | NVARCHAR2(512) | nvarchar(512) | ID of the associated ThinApp™ object. |
| Time | TIMESTAMP | datetime | Time at which the event occurred, measured from the epoch (January 1, 1970). |
| UserDiskPathId | NVARCHAR2(512) | nvarchar(512) | ID of the user disk. |
| UserSID | NVARCHAR2(512) | nvarchar(512) | SID of the associated user in Active Directory. |

Table 2-3 shows the schema for the event_data and event_data_historical tables.

**Table 2-3.** Schema for the event_data and event_data_historical Tables

| Column Name | Oracle Data Type | SQLServer Data Type | Description |
|---|---|---|---|
| BooleanValue | SMALLINT | tinyint | Value of a Boolean attribute.<br>0 = false<br>1 = true |
| EventID | INTEGER | int | Unique primary key for the event. |
| IntValue | INTEGER | int | Value of an integer attribute. |
| Name | NVARCHAR2(512) | nvarchar(512) | Attribute name (for example, UserDisplayName). |
| StrValue | NVARCHAR2(512) | nvarchar(512) | Value of a string attribute. For other types of attributes, this column contains an interpretation of the data type as a string. |
| TimeValue | TIMESTAMP | datetime | Value of a date and time attribute. |
| Type | SMALLINT | tinyint | The data type of the attribute.<br>0 = StrValue<br>1 = IntValue<br>2 = TimeValue<br>3 = BooleanValue |

# Connection Broker Events

Table 2-4 shows the event types for the connection broker.

The BROKER_DAILY_MAX_DESKTOP_SESSIONS event reports the maximum number of concurrent desktop sessions over a 24-hour period. If a user runs multiple desktop sessions concurrently, each desktop session is counted separately. The BROKER_DAILY_MAX_APP_USERS event reports the maximum number of concurrent application users over a 24-hour period. If a user runs multiple applications concurrently, the user is counted only once. Short-lived sessions might not be included in the count because the sampling is performed every five minutes.

The BROKER_VC_DISABLED and BROKER_VC_ENABLED events report the state of of the vCenter driver that View uses to track a vCenter Server. The BROKER_VC_STATUS_* events report the state of a vCenter Server.

**Table 2-4.** Connection Broker Events

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| BROKER_AGENT_OFFLINE | WARNING | The agent running on machine ${MachineName} has not responded to queries, marking it as offline |
| BROKER_AGENT_ONLINE | WARNING | The agent running on machine ${MachineName} is responding again, but did not send a startup message |
| BROKER_APPLICATION_LAUNCH_FAILURE | ERROR | Unable to launch from Pool ${PoolId} for user ${UserDisplayName}: The broker encountered an error while processing the request, please contact support for assistance. |
| BROKER_APPLICATION_MISSING | WARNING | At least ${ApplicationMissingCount} applications, including ${ApplicationExecutable}, are not installed on ${MachineName} in Pool ${PoolId} |

**Table 2-4.** Connection Broker Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| BROKER_APPLICATION_NOT_ENTITLED | AUDIT_FAIL | Unable to launch from Pool ${PoolId} for user ${UserDisplayName}: User is not entitled to this Pool |
| BROKER_APPLICATION_PROTOCOL_NOT_ SUPPORTED | AUDIT_FAIL | Unable to launch from Pool ${PoolId} for user ${UserDisplayName}: Requested protocol ${ProtocolId} is not supported |
| BROKER_APPLICATION_REQUEST | INFO | User ${UserDisplayName} requested Application ${ApplicationId} |
| BROKER_APPLICATION_SESSION_REQUEST | INFO | User ${UserDisplayName} requested an application session from Pool ${PoolId} |
| BROKER_DAILY_MAX_DESKTOP_SESSIONS | INFO | ${Time}: Over the past 24 hours, the maximum number of concurrent desktop sessions was ${UserCount}. |
| BROKER_DAILY_MAX_APP_USERS | INFO | ${Time}: Over the past 24 hours, the maximum number of users with concurrent application sessions was ${UserCount}. |
| BROKER_DESKTOP_LAUNCH_FAILURE | ERROR | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: The broker encountered an error while processing the request, please contact support for assistance |
| BROKER_DESKTOP_NOT_ENTITLED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: User is not entitled to this Pool |
| BROKER_DESKTOP_PROTOCOL_NOT_ SUPPORTED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Requested protocol ${ProtocolId} is not supported |
| BROKER_DESKTOP_REQUEST | INFO | User ${UserDisplayName} requested Pool ${DesktopId} |
| BROKER_EVENT_HANDLING_STARTED | INFO | Broker ${BrokerName} has started handling events |
| BROKER_EVENT_HANDLING_STOPPED | INFO | ${BrokerName} has stopped handling events |
| BROKER_MACHINE_ALLOCATED | INFO | User ${UserDisplayName} requested Pool ${DesktopId}, allocated machine ${MachineName} |
| BROKER_MACHINE_ASSIGNED_ UNAVAILABLE | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Assigned machine ${MachineName} is unavailable |
| BROKER_MACHINE_CANNOT_CONNECT | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Failed to connect to Machine ${MachineName} using ${ProtocolId} |
| BROKER_MACHINE_CONFIGURED_VIDEO_ SETTINGS | INFO | Successfully configured video settings for Machine VM ${MachineName} in Pool ${DesktopId} |
| BROKER_MACHINE_NOT_READY | WARNING | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Machine ${MachineName} is not ready to accept connections |
| BROKER_MACHINE_OPERATION_DELETED | INFO | machine ${MachineName} has been deleted |

**Table 2-4.** Connection Broker Events (Continued)

| EventType | Severity | ModuleAndEventText |
| --- | --- | --- |
| BROKER_MACHINE_PROTOCOL_NOT_SUPPORTED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Machine ${MachineName} does not support protocol ${ProtocolId} |
| BROKER_MACHINE_PROTOCOL_UNAVAILABLE | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Machine ${MachineName} did not report protocol ${ProtocolId} as ready |
| BROKER_MACHINE_REJECTED_SESSION | WARNING | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Machine ${MachineName} rejected the start session request |
| BROKER_MACHINE_SESSION_TIMEDOUT | WARNING | Session for user ${UserDisplayName} timed out |
| BROKER_MULTIPLE_DESKTOPS_FOR_KIOSK_USER | WARNING | User ${UserDisplayName} is entitled to multiple desktop pools |
| BROKER_POOL_CANNOT_ASSIGN | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: There are no machines available to assign the user to |
| BROKER_POOL_COMANAGER_REQUIRED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: No co-management availability for protocol ${ProtocolId} |
| BROKER_POOL_EMPTY | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: The Desktop Pool is empty |
| BROKER_POOL_NO_MACHINE_ASSIGNED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: No machine assigned to this user |
| BROKER_POOL_NO_RESPONSES | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: No machines in the Desktop Pool are responsive |
| BROKER_POOL_OVERLOADED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: All responding machines are currently in use |
| BROKER_POOL_POLICY_VIOLATION | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: This Desktop Pool does not allow online sessions |
| BROKER_POOL_PROTOCOL_NOT_SUPPORTED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: There were no machines available that support protocol ${ProtocolId} |
| BROKER_POOL_PROTOCOL_UNAVAILABLE | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: There were no machines available that reported protocol ${ProtocolId} as ready |
| BROKER_POOL_TUNNEL_NOT_SUPPORTED | AUDIT_FAIL | Unable to launch from Pool ${DesktopId} for user ${UserDisplayName}: Tunnelling is not supported for protocol ${ProtocolId} |
| BROKER_PROVISIONING_ERROR_CONFIG_CLEARED | INFO | The previously reported configuration problem is no longer present on Pool ${DesktopId} |

**Table 2-4.** Connection Broker Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| BROKER_PROVISIONING_ERROR_CONFIG_ SET | ERROR | Provisioning error occurred on Pool ${DesktopId} because of a configuration problem |
| BROKER_PROVISIONING_ERROR_DISK_ CLEARED | INFO | The previously reported disk problem is no longer present on Pool ${DesktopId} |
| BROKER_PROVISIONING_ERROR_DISK_LC_ RESERVATION_CLEARED | INFO | The previously reported error due to available free disk space reserved for linked clones is no longer present on Pool ${DesktopId} |
| BROKER_PROVISIONING_ERROR_DISK_LC_ RESERVATION_SET | ERROR | Provisioning error occurred on Pool ${DesktopId} because available free disk space is reserved for linked clones |
| BROKER_PROVISIONING_ERROR_DISK_SET | WARNING | Provisioning error occurred on Pool ${DesktopId} because of a disk problem |
| BROKER_PROVISIONING_ERROR_LICENCE_ CLEARED | INFO | The previously reported licensing problem is no longer present on Pool ${DesktopId} |
| BROKER_PROVISIONING_ERROR_LICENCE_ SET | ERROR | Provisioning error occurred on Pool ${DesktopId} because of a licensing problem |
| BROKER_PROVISIONING_ERROR_ NETWORKING_CLEARED | INFO | The previously reported networking problems with a View Agent are no longer present on Pool ${DesktopId} |
| BROKER_PROVISIONING_ERROR_ NETWORKING_SET | ERROR | Provisioning error occurred on Pool ${DesktopId} because of a networking problem with a View Agent |
| BROKER_PROVISIONING_ERROR_RESOURCE_ CLEARED | INFO | The previously reported resource problem is no longer present on Pool ${DesktopId} |
| BROKER_PROVISIONING_ERROR_RESOURCE_ SET | ERROR | Provisioning error occurred on Pool ${DesktopId} because of a resource problem |
| BROKER_PROVISIONING_ERROR_TIMEOUT_ CUSTOMIZATION_CLEARED | INFO | The previously reported timeout while customizing is no longer present on Pool ${DesktopId} |
| BROKER_PROVISIONING_ERROR_TIMEOUT_ CUSTOMIZATION_SET | ERROR | Provisioning error occurred on Pool ${DesktopId} because of a timeout while customizing |
| BROKER_PROVISIONING_ERROR_VM_ CLONING | ERROR | Provisioning error occurred for Machine ${MachineName}: Cloning failed for Machine |
| BROKER_PROVISIONING_ERROR_VM_ CUSTOMIZATION_ERROR | ERROR | Provisioning error occurred for Machine ${MachineName}: Customization failed for Machine |
| BROKER_PROVISIONING_ERROR_VM_ CUSTOMIZATION_NETWORKING | ERROR | Provisioning error occurred for Machine ${MachineName}: Customization error due to no network communication between the View agent and Connection Server |
| BROKER_PROVISIONING_ERROR_VM_ CUSTOMIZATION_TIMEOUT | ERROR | Provisioning error occurred for Machine ${MachineName}: Customization operation timed out |
| BROKER_PROVISIONING_SVI_ERROR_ COMPOSER_AGENT_INIT_FAILED | ERROR | Provisioning error occurred for Machine ${MachineName}: View Composer agent initialization failed |

**Table 2-4.**  Connection Broker Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| BROKER_PROVISIONING_SVI_ERROR_ RECONFIG_FAILED | ERROR | Provisioning error occurred for Machine ${MachineName}: Reconfigure operation failed |
| BROKER_PROVISIONING_SVI_ERROR_REFIT_ FAILED | ERROR | Provisioning error occurred for Machine ${MachineName}: Refit operation ${SVIOperation} failed |
| BROKER_PROVISIONING_SVI_ERROR_ REMOVING_VM | ERROR | Provisioning error occurred for Machine ${MachineName}: Unable to remove Machine from inventory |
| BROKER_PROVISIONING_VERIFICATION_ FAILED_USER_ASSIGNED | WARNING | Provisioning verification failed for Machine ${MachineName}: User is already assigned to a machine in Pool ${DesktopId} |
| BROKER_PROVISIONING_VERIFICATION_ FAILED_USER_CANNOT_BE_ASSIGNED | WARNING | Provisioning verification failed for Machine ${MachineName}: A user cannot be assigned because Pool ${DesktopId} is not persistent |
| BROKER_PROVISIONING_VERIFICATION_ FAILED_VMNAME_IN_USE | WARNING | Provisioning verification failed for Machine ${MachineName}: A machine already exists in Pool ${DesktopId} with name ${MachineName} |
| BROKER_SECURITY_SERVER_ADD_FAILED | AUDIT_FAIL | Failed to add security server ${SecurityServerId} |
| BROKER_SECURITY_SERVER_ADD_FAILED_ PASSWORD_EXPIRED | AUDIT_FAIL | Failed to add security server ${SecurityServerId}, pairing password expired |
| BROKER_SECURITY_SERVER_ADD_FAILED_ PASSWORD_INCORRECT | AUDIT_FAIL | Failed to add security server ${SecurityServerId}, pairing password incorrect |
| BROKER_SECURITY_SERVER_ADD_FAILED_ PASSWORD_NOT_SET | AUDIT_FAIL | Failed to add security server ${SecurityServerId}, pairing password not set |
| BROKER_SECURITY_SERVER_ADDED | AUDIT_SUCCESS | Security server ${SecurityServerId} added |
| BROKER_SVI_ARCHIVE_UDD_FAILED | AUDIT_FAIL | Failed to archive user data disk ${UserDiskName} to location ${SVIPath} |
| BROKER_SVI_ARCHIVE_UDD_SUCCEEDED | AUDIT_SUCCESS | Archived user data disk ${UserDiskName} to location ${SVIPath} |
| BROKER_SVI_ATTACH_UDD_FAILED | AUDIT_FAIL | Failed to attach user data disk ${UserDiskName} to VM ${SVIVMID} |
| BROKER_SVI_ATTACH_UDD_SUCCEEDED | AUDIT_SUCCESS | Attached user data disk ${UserDiskName} to VM ${SVIVMID} |
| BROKER_SVI_DETACH_UDD_FAILED | AUDIT_FAIL | Failed to detach user data disk ${UserDiskName} from VM ${SVIVMID} |
| BROKER_SVI_DETACH_UDD_SUCCEEDED | AUDIT_SUCCESS | Detached user data disk ${UserDiskName} from VM ${SVIVMID} |
| BROKER_USER_AUTHFAILED_ACCOUNT_ DISABLED | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because the account is disabled |
| BROKER_USER_AUTHFAILED_ACCOUNT_ EXPIRED | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because the account has expired |
| BROKER_USER_AUTHFAILED_ACCOUNT_ LOCKED_OUT | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because the account is locked out |

**Table 2-4.** Connection Broker Events (Continued)

| EventType | Severity | ModuleAndEventText |
| --- | --- | --- |
| BROKER_USER_AUTHFAILED_ACCOUNT_RESTRICTION | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because of an account restriction |
| BROKER_USER_AUTHFAILED_BAD_USER_PASSWORD | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because of a bad username or password |
| BROKER_USER_AUTHFAILED_GENERAL | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate |
| BROKER_USER_AUTHFAILED_NO_LOGON_SERVERS | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because there are no logon servers |
| BROKER_USER_AUTHFAILED_PASSWORD_EXPIRED | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because the password has expired |
| BROKER_USER_AUTHFAILED_PASSWORD_MUST_CHANGE | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because the password must change |
| BROKER_USER_AUTHFAILED_SECUREID_ACCESS_DENIED | AUDIT_FAIL | SecurID access denied for user ${UserDisplayName} |
| BROKER_USER_AUTHFAILED_SECUREID_NEWPIN_REJECTED | AUDIT_FAIL | SecurID access denied for user ${UserDisplayName} because new pin was rejected |
| BROKER_USER_AUTHFAILED_SECUREID_WRONG_NEXTTOKEN | AUDIT_FAIL | SecurID access denied for user ${UserDisplayName} because wrong next token entered |
| BROKER_USER_AUTHFAILED_SECUREID_WRONG_STATE | AUDIT_FAIL | SecurID access denied for user ${UserDisplayName} because of incorrect state |
| BROKER_USER_AUTHFAILED_TIME_RESTRICTION | AUDIT_FAIL | User ${UserDisplayName} failed to authenticate because of a time restriction |
| BROKER_USER_NOT_AUTHORIZED | AUDIT_FAIL | User ${UserDisplayName} has authenticated, but is not authorized to perform the operation |
| BROKER_USER_NOT_ENTITLED | AUDIT_FAIL | User ${UserDisplayName} has authenticated, but is not entitled to any Pools |
| BROKER_USERCHANGEDPASSWORD | AUDIT_SUCCESS | Password for ${UserDisplayName} has been changed by the user |
| BROKER_USERLOGGEDIN | AUDIT_SUCCESS | User ${UserDisplayName} has logged in |
| BROKER_USERLOGGEDOUT | AUDIT_SUCCESS | User ${UserDisplayName} has logged out |
| BROKER_VC_DISABLED | INFO | vCenter at address ${VCAddress} has been temporarily disabled |
| BROKER_VC_ENABLED | INFO | vCenter at address ${VCAddress} has been enabled |
| BROKER_VC_STATUS_CHANGED_CANNOT_LOGIN | WARNING | Cannot log in to vCenter at address ${VCAddress} |
| BROKER_VC_STATUS_CHANGED_DOWN | INFO | vCenter at address ${VCAddress} is down |
| BROKER_VC_STATUS_CHANGED_INVALID_CREDENTIALS | WARNING | vCenter at address ${VCAddress} has invalid credentials |
| BROKER_VC_STATUS_CHANGED_NOT_YET_CONNECTED | INFO | Not yet connected to vCenter at address ${VCAddress} |

**Table 2-4.**  Connection Broker Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| BROKER_VC_STATUS_CHANGED_ RECONNECTING | INFO | Reconnecting to vCenter at address ${VCAddress} |
| BROKER_VC_STATUS_CHANGED_UNKNOWN | WARNING | The status of vCenter at address ${VCAddress} is unknown |
| BROKER_VC_STATUS_CHANGED_UP | INFO | vCenter at address ${VCAddress} is up |

# View Agent Events

Table 2-5 shows the event types for View Agent.

**Table 2-5.**  View Agent Events

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| AGENT_CONNECTED | INFO | User ${UserDisplayName} has logged in to a new session on machine ${MachineName} |
| AGENT_DISCONNECTED | INFO | User ${UserDisplayName} has disconnected from machine ${MachineName} |
| AGENT_ENDED | INFO | User ${UserDisplayName} has logged off machine ${MachineName} |
| AGENT_PENDING | INFO | The agent running on machine ${MachineName} has accepted an allocated session for user ${UserDisplayName} |
| AGENT_PENDING_ EXPIRED | WARNING | The pending session on machine ${MachineName} for user ${UserDisplayName} has expired |
| AGENT_RECONFIGURED | INFO | Machine ${MachineName} has been successfully reconfigured |
| AGENT_RECONNECTED | INFO | User ${UserDisplayName} has reconnected to machine ${MachineName} |
| AGENT_RESUME | INFO | The agent on machine ${MachineName} sent a resume message |
| AGENT_SHUTDOWN | INFO | The agent running on machine ${MachineName} has shut down, this machine will be unavailable |
| AGENT_STARTUP | INFO | The agent running on machine ${MachineName} has contacted the connection server and sent a startup message |
| AGENT_SUSPEND | INFO | The agent on machine ${MachineName} sent a suspend message |

# View Administrator Events

Table 2-6 shows the event types for View Administrator.

**Table 2-6.**  View Administrator Events

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| ADMIN_ADD_DESKTOP_ENTITLEMENT | AUDIT_SUCCESS | ${EntitlementDisplay} was entitled to Pool ${DesktopId} by ${UserDisplayName} |
| ADMIN_ADD_LICENSE | AUDIT_SUCCESS | ${UserDisplayName} added license |
| ADMIN_ADD_LICENSE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add license |
| ADMIN_ADD_PM | AUDIT_SUCCESS | ${UserDisplayName} added physical machine ${MachineName} to Pool ${DesktopId} |
| ADMIN_ADD_PM_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add physical machine ${MachineName} to Pool ${DesktopId} |

**Table 2-6.** View Administrator Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| ADMIN_ADD_THINAPP_ENTITLEMENT | AUDIT_SUCCESS | Application ${ThinAppDisplayName} was assigned to Desktop ${MachineName} by ${UserDisplayName} |
| ADMIN_ADD_THINAPP_ENTITLEMENT_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add Application entitlement |
| ADMIN_ADD_THINAPP_POOL_ ENTITLEMENT | AUDIT_SUCCESS | Application ${ThinAppDisplayName} was assigned to Pool ${DesktopId} by ${UserDisplayName} |
| ADMIN_ADMINSTRATOR_REMOVE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove all permissions for Administrator ${AdminPermissionEntity} |
| ADMIN_ADMINSTRATOR_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed all permissions for Administrator ${AdminPermissionEntity} |
| ADMIN_CONNECTION_BROKER_UPDATE_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update connection broker ${BrokerId} |
| ADMIN_CONNECTION_BROKER_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated connection broker ${BrokerId}: (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_CONNECTION_SERVER_BACKUP_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to initiate a backup of connection broker ${BrokerId} |
| ADMIN_CONNECTION_SERVER_BACKUP_ INITIATED | AUDIT_SUCCESS | ${UserDisplayName} initiated a backup of connection broker ${BrokerId} |
| ADMIN_CONNECTION_SERVER_DISABLE_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to disable connection broker ${BrokerId} |
| ADMIN_CONNECTION_SERVER_DISABLED | AUDIT_SUCCESS | ${UserDisplayName} is disabling connection broker ${BrokerId} |
| ADMIN_CONNECTION_SERVER_ENABLE_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to enable connection broker ${BrokerId} |
| ADMIN_CONNECTION_SERVER_ENABLED | AUDIT_SUCCESS | ${UserDisplayName} is enabling connection broker ${BrokerId} |
| ADMIN_DATABASE_CONFIGURATION_ADD_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add database configuration |
| ADMIN_DATABASE_CONFIGURATION_ ADDED | AUDIT_SUCCESS | ${UserDisplayName} has added database configuration |
| ADMIN_DATABASE_CONFIGURATION_ DELETE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to delete database configuration |
| ADMIN_DATABASE_CONFIGURATION_ DELETE_FAILED | AUDIT_SUCCESS | ${UserDisplayName} has deleted database configuration |
| ADMIN_DATABASE_CONFIGURATION_ UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update database configuration |
| ADMIN_DATABASE_CONFIGURATION_ UPDATED | AUDIT_SUCCESS | ${UserDisplayName} has updated database configuration |
| ADMIN_DEFAULT_DESKTOPPOOL_ASSIGN | AUDIT_SUCCESS | ${UserDisplayName} assigned Pool ${DesktopId} for default desktop to ${UserName} |
| ADMIN_DEFAULT_DESKTOPPOOL_ASSIGN_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to assign Pool ${DesktopId} for default desktop to ${UserName} |
| ADMIN_DEFAULT_DESKTOPPOOL_ UNASSIGN | AUDIT_SUCCESS | ${UserDisplayName} removed pool assignment for default desktop to ${UserName} |

**Table 2-6.** View Administrator Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| ADMIN_DEFAULT_DESKTOPPOOL_UNASSIGN_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove Pool assignment for default desktop to ${UserName} |
| ADMIN_DESKTOP_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added Pool ${DesktopId} |
| ADMIN_DESKTOP_ASSIGN | AUDIT_SUCCESS | ${UserDisplayName} assigned Desktop ${MachineName} to ${UserName} |
| ADMIN_DESKTOP_ASSIGN_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to assign Desktop ${MachineName} to ${UserName} |
| ADMIN_DESKTOP_EDITED | AUDIT_SUCCESS | ${UserDisplayName} edited Pool ${DesktopId} (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_DESKTOP_MAINTENANCE_MODE_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update desktop ${MachineName} to ${MaintenanceMode} maintenance mode |
| ADMIN_DESKTOP_MAINTENANCE_MODE_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated desktop ${MachineName} to ${MaintenanceMode} maintenance mode |
| ADMIN_DESKTOP_UNASSIGN | AUDIT_SUCCESS | ${UserDisplayName} removed assignment for Desktop ${MachineName} |
| ADMIN_DESKTOP_UNASSIGN_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove assignment for Desktop ${MachineName} |
| ADMIN_ENABLE_DESKTOP_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to set Pool ${DesktopId} to ${EnableStatus} |
| ADMIN_ENABLE_DESKTOP_SUCCEEDED | AUDIT_SUCCESS | ${UserDisplayName} set Pool ${DesktopId} to ${EnableStatus} |
| ADMIN_ENABLED_DESKTOP_PROVISION_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to set provisioning for Pool ${DesktopId} to ${EnableStatus} |
| ADMIN_ENABLED_DESKTOP_PROVISION_SUCCEEDED | AUDIT_SUCCESS | ${UserDisplayName} set provisioning for Pool ${DesktopId} to ${EnableStatus} |
| ADMIN_EVENT_CONFIGURATION_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update event configuration |
| ADMIN_EVENT_CONFIGURATION_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} has updated global configuration |
| ADMIN_FOLDER_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add folder ${AdminFolderName} |
| ADMIN_FOLDER_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added folder ${AdminFolderName} |
| ADMIN_FOLDER_CHANGE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to change object ${ObjectID}(type=${ObjectType}) to folder ${AdminFolderName} |
| ADMIN_FOLDER_CHANGED | AUDIT_SUCCESS | ${UserDisplayName} changed object ${ObjectID}(type=${ObjectType}) to folder ${AdminFolderName} |
| ADMIN_FOLDER_DELETE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to delete folder ${AdminFolderName} |
| ADMIN_FOLDER_DELETED | AUDIT_SUCCESS | ${UserDisplayName} deleted folder ${AdminFolderName} |
| ADMIN_GLOBAL_CONFIGURATION_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update global configuration |

**Table 2-6.** View Administrator Events (Continued)

| EventType | Severity | ModuleAndEventText |
| --- | --- | --- |
| ADMIN_GLOBAL_CONFIGURATION_ UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated global configuration (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_GLOBAL_POLICY_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update global policies |
| ADMIN_GLOBAL_POLICY_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated global policy (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_PERFMON_CONFIGURATION_ UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update performance monitoring configuration |
| ADMIN_PERFMON_CONFIGURATION_ UPDATED | AUDIT_SUCCESS | ${UserDisplayName} has updated performance monitoring configuration |
| ADMIN_PERMISSION_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add Permission to ${AdminPermissionEntity} with Role ${AdminRoleName} on Folder ${AdminFolderName} |
| ADMIN_PERMISSION_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added Permission to ${AdminPermissionEntity} with Role ${AdminRoleName} on Folder ${AdminFolderName} |
| ADMIN_PERMISSION_REMOVE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove Permission to ${AdminPermissionEntity} with Role ${AdminRoleName} on Folder ${AdminFolderName} |
| ADMIN_PERMISSION_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed Permission to ${AdminPermissionEntity} with Role ${AdminRoleName} on Folder ${AdminFolderName} |
| ADMIN_POOL_POLICY_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update Pool ${DesktopId} policies |
| ADMIN_POOL_POLICY_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated Pool ${DesktopId} policy (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_REMOVE_DESKTOP_ENTITLEMENT | AUDIT_SUCCESS | ${EntitlementDisplay} was unentitled from Pool ${DesktopId} by ${UserDisplayName} |
| ADMIN_REMOVE_DESKTOP_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to removed Pool ${DesktopId} |
| ADMIN_REMOVE_DESKTOP_SUCCEEDED | AUDIT_SUCCESS | ${UserDisplayName} removed Pool ${DesktopId} |
| ADMIN_REMOVE_THINAPP_ENTITLEMENT | AUDIT_SUCCESS | Application ${ThinAppDisplayName} was unassigned from Desktop ${MachineName} by ${UserDisplayName} |
| ADMIN_REMOVE_THINAPP_ENTITLEMENT_ FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove Application entitlement |
| ADMIN_REMOVE_THINAPP_POOL_ ENTITLEMENT | AUDIT_SUCCESS | Application ${ThinAppDisplayName} was unassigned from Pool ${DesktopId} by ${UserDisplayName} |
| ADMIN_RESET_THINAPP_STATE | AUDIT_SUCCESS | Application ${ThinAppDisplayName} state are reset for Desktop ${DesktopDisplayName} by ${UserDisplayName} |
| ADMIN_RESET_THINAPP_STATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to reset Application state for ${ThinAppDisplayName} |

**Table 2-6.** View Administrator Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| ADMIN_ROLE_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add Role ${AdminRoleName} with privileges ${AdminPrivilegeName} |
| ADMIN_ROLE_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added Role ${AdminRoleName} with privileges ${AdminPrivilegeName} |
| ADMIN_ROLE_PRIV_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update Role ${AdminRoleName} to privileges ${AdminPrivilegeName} |
| ADMIN_ROLE_PRIV_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated Role ${AdminRoleName} to privileges ${AdminPrivilegeName} |
| ADMIN_ROLE_REMOVE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove Role ${AdminRoleName} |
| ADMIN_ROLE_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed Role ${AdminRoleName} |
| ADMIN_ROLE_RENAME_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to rename Role ${AdminRoleName} to ${AdminRoleNewName} |
| ADMIN_ROLE_RENAMED | AUDIT_SUCCESS | ${UserDisplayName} renamed Role ${AdminRoleName} to ${AdminRoleNewName} |
| ADMIN_SECURITY_SERVER_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add security server ${SecurityServerId} |
| ADMIN_SECURITY_SERVER_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added security server ${SecurityServerId} |
| ADMIN_SECURITY_SERVER_EDIT_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to edit security server ${SecurityServerId} |
| ADMIN_SECURITY_SERVER_EDITED | AUDIT_SUCCESS | ${UserDisplayName} edited security server ${SecurityServerId} (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_SECURITY_SERVER_REMOVE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove security server ${SecurityServerId} |
| ADMIN_SECURITY_SERVER_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed security server ${SecurityServerId} |
| ADMIN_SESSION_SENDMSG | AUDIT_SUCCESS | ${UserDisplayName} sent message (${SessionMessage}) to session (User ${UserName}, Desktop ${MachineName}) |
| ADMIN_SESSION_SENDMSG_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to send message (${SessionMessage}) to session ${ObjectId} |
| ADMIN_SVI_ADD_DEPLOYMENT_GROUP_FAILED | AUDIT_FAIL | Failed to add deployment group for ${SVIParentVM} : ${SVISnapshot} |
| ADMIN_SVI_ADD_DEPLOYMENT_GROUP_SUCCEEDED | AUDIT_SUCCESS | Added deployment group ${SVIDeploymentGroupID} for ${SVIParentVM} : ${SVISnapshot} |
| ADMIN_SVI_ADD_UDD_FAILED | AUDIT_FAIL | Failed to add user data disk ${UserDiskName} |
| ADMIN_SVI_ADD_UDD_SUCCEEDED | AUDIT_SUCCESS | Added user data disk ${UserDiskName} |
| ADMIN_SVI_ADMIN_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added SVI QuickPrep domain ${SVIAdminFqdn}(${SVIAdminName}) |

**Table 2-6.** View Administrator Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| ADMIN_SVI_ADMIN_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed SVI QuickPrep domain (id=${SVIAdminID}) |
| ADMIN_SVI_ADMIN_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated SVI QuickPrep domain ${SVIAdminFqdn}(${SVIAdminName}) |
| ADMIN_SVI_ATTACH_UDD_FAILED | AUDIT_FAIL | Failed to request attach user data disk ${UserDiskName} to VM ${SVIVMID} |
| ADMIN_SVI_ATTACH_UDD_SUCCEEDED | AUDIT_SUCCESS | Requested attach user data disk ${UserDiskName} to VM ${SVIVMID} |
| ADMIN_SVI_DELETE_UDD_FAILED | AUDIT_FAIL | Failed to delete user data disk ${UserDiskName} |
| ADMIN_SVI_DELETE_UDD_SUCCEEDED | AUDIT_SUCCESS | Deleted user data disk ${UserDiskName} |
| ADMIN_SVI_DETACH_UDD_FAILED | AUDIT_FAIL | Failed to request detach user data disk ${UserDiskName} from VM ${SVIVMID} |
| ADMIN_SVI_DETACH_UDD_SUCCEEDED | AUDIT_SUCCESS | Requested detach user data disk ${UserDiskName} from VM ${SVIVMID} |
| ADMIN_SVI_REBALANCE_VM_FAILED | AUDIT_FAIL | Failed to rebalance VM ${SVIVMID} |
| ADMIN_SVI_REBALANCE_VM_SUCCEEDED | AUDIT_SUCCESS | Rebalanced VM ${SVIVMID} |
| ADMIN_SVI_REFRESH_VM_FAILED | AUDIT_FAIL | Failed to refresh VM ${SVIVMID} |
| ADMIN_SVI_REFRESH_VM_SUCCEEDED | AUDIT_SUCCESS | Refreshed VM ${SVIVMID} |
| ADMIN_SVI_RESYNC_VM_FAILED | AUDIT_FAIL | Failed to resync VM ${SVIVMID} to deployment group ${SVIDeploymentGroupID} |
| ADMIN_SVI_RESYNC_VM_SUCCEEDED | AUDIT_SUCCESS | Resyncd VM ${SVIVMID} to deployment group ${SVIDeploymentGroupID} |
| ADMIN_SVI_UPDATE_POOL_DEPLOYMENT_GROUP_FAILED | AUDIT_FAIL | Failed to update pool ${DesktopId} to deployment group ${SVIDeploymentGroupID} |
| ADMIN_SVI_UPDATE_POOL_DEPLOYMENT_GROUP_SUCCEEDED | AUDIT_SUCCESS | Updated pool ${DesktopId} to deployment group ${SVIDeploymentGroupID} |
| ADMIN_SVI_UPDATE_UDD_FAILED | AUDIT_FAIL | Failed to update user data disk ${UserDiskName} |
| ADMIN_SVI_UPDATE_UDD_SUCCEEDED | AUDIT_SUCCESS | Set user data disk ${UserDiskName} pool to ${DesktopId} and user to ${UserName} |
| ADMIN_THINAPP_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add Application ${ThinAppDisplayName} |
| ADMIN_THINAPP_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added Application ${ThinAppDisplayName} |
| ADMIN_THINAPP_DESKTOP_AVAILABLE | AUDIT_SUCCESS | Application ${ThinAppDisplayName} is now available on Desktop ${DesktopDisplayName} |
| ADMIN_THINAPP_DESKTOP_REMOVED | AUDIT_SUCCESS | Application ${ThinAppDisplayName} has been removed from Desktop ${DesktopDisplayName} |
| ADMIN_THINAPP_EDITED | AUDIT_SUCCESS | ${UserDisplayName} edited Application ${ThinAppDisplayName} |
| ADMIN_THINAPP_FAILED_DESKTOP_DELIVERY | AUDIT_FAIL | Failed to deliver Application ${ThinAppDisplayName} to Desktop ${DesktopDisplayName} |

**Table 2-6.**  View Administrator Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| ADMIN_THINAPP_FAILED_DESKTOP_ REMOVAL | AUDIT_FAIL | Failed to remove Application ${ThinAppDisplayName} from Desktop ${DesktopDisplayName} |
| ADMIN_THINAPP_GROUP_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add Application Template ${ThinAppGroupName} |
| ADMIN_THINAPP_GROUP_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added Application Template ${ThinAppGroupName} with Applications ${ThinAppGroupApplications} |
| ADMIN_THINAPP_GROUP_EDIT_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to edit Application Template ${ThinAppGroupName} |
| ADMIN_THINAPP_GROUP_EDITED | AUDIT_SUCCESS | ${UserDisplayName} edited Application Template ${ThinAppGroupName} with Applications ${ThinAppGroupApplications} |
| ADMIN_THINAPP_GROUP_REMOVE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove Application Template ${ThinAppGroupName} |
| ADMIN_THINAPP_GROUP_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed Application Template ${ThinAppGroupName} |
| ADMIN_THINAPP_REMOVE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove Application ${ThinAppDisplayName} |
| ADMIN_THINAPP_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed Application ${ThinAppDisplayName} |
| ADMIN_THINAPP_REPO_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add Repository ${ThinAppRepositoryName}, path ${ThinAppRepositoryPath} |
| ADMIN_THINAPP_REPO_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added Repository ${ThinAppRepositoryName}, path ${ThinAppRepositoryPath} |
| ADMIN_THINAPP_REPO_EDIT_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to edit Repository ${ThinAppRepositoryName}, path ${ThinAppRepositoryPath} |
| ADMIN_THINAPP_REPO_EDITED | AUDIT_SUCCESS | ${UserDisplayName} edited Repository ${ThinAppRepositoryName}, path ${ThinAppRepositoryPath} |
| ADMIN_THINAPP_REPO_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed Repository ${ThinAppRepositoryName} |
| ADMIN_UNREGISTER_PM | AUDIT_SUCCESS | ${UserDisplayName} unregistered physical machine ${MachineName}) |
| ADMIN_UNREGISTER_PM_FAILED | AUDIT_FAIL | ${UserDisplayName} fails to unregister physical machine ${MachineName}) |
| ADMIN_USER_INFO_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update user info with AD server for ${UserName} |
| ADMIN_USER_INFO_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated user info with AD server for ${UserName} |
| ADMIN_USER_POLICY_DELETE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to delete Pool ${DesktopId} override policies for user ${UserName} |

**Table 2-6.** View Administrator Events (Continued)

| EventType | Severity | ModuleAndEventText |
|---|---|---|
| ADMIN_USER_POLICY_DELETED | AUDIT_SUCCESS | ${UserDisplayName} deleted Pool ${DesktopId} override policy for user ${UserName} (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_USER_POLICY_UPDATE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to update Pool ${DesktopId} policies for user ${UserName} |
| ADMIN_USER_POLICY_UPDATED | AUDIT_SUCCESS | ${UserDisplayName} updated Pool ${DesktopId} policy for user ${UserName} (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_USERLOGGEDIN | AUDIT_SUCCESS | User ${UserDisplayName} has logged in to View Administrator |
| ADMIN_USERLOGGEDOUT | AUDIT_SUCCESS | User ${UserDisplayName} has logged out from View Administrator |
| ADMIN_VC_ADD_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to add VC server ${VCAddress} |
| ADMIN_VC_ADDED | AUDIT_SUCCESS | ${UserDisplayName} added VC server ${VCAddress} |
| ADMIN_VC_EDITED | AUDIT_SUCCESS | ${UserDisplayName} edited VC server ${VCAddress} (${AttrChangeType}: ${AttrName} = ${AttrValue}) |
| ADMIN_VC_LICINV_ALARM_DISABLED | AUDIT_SUCCESS | Alarm on VC server ${VCAddress} for License Inventory monitoring was disabled as all Hosts have desktop licenses |
| ADMIN_VC_REMOVE_FAILED | AUDIT_FAIL | ${UserDisplayName} failed to remove VC server ${VCAddress} |
| ADMIN_VC_REMOVED | AUDIT_SUCCESS | ${UserDisplayName} removed VC server ${VCAddress} |

## Event Message Attributes

Table 2-7 shows the attributes that ModuleAndEventText messages use. To determine the data type of an attribute, you can examine the value of the Type column in the event_data or event_data_historical table.

**Table 2-7.** Attributes Used with ModuleAndEventText Messages

| Attribute Name | Description |
|---|---|
| AdminFolderName | Name of a folder requiring privileged access. |
| AdminPermissionEntity | Name of an object requiring privileged access. |
| AdminPrivilegeName | Name of an administrative privilege. |
| AdminRoleName | Name of an administrative role. |
| AdminRoleNewName | New name of an administrative role. |
| AttrChangeType | Type of change that was applied to a generic attribute. |
| AttrName | Name of a generic attribute. |
| AttrValue | Value of a generic attribute. |
| BrokerId | Identifier of a View Connection Server instance. |
| BrokerName | Name of a View Connection Server instance. |
| DesktopDisplayName | Display name of a desktop pool. |
| DesktopId | Identifier of a desktop pool. |

**Table 2-7.** Attributes Used with ModuleAndEventText Messages (Continued)

| Attribute Name | Description |
| --- | --- |
| EntitlementDisplay | Display name of a desktop entitlement. |
| MachineId | Identifier of a physical or virtual machine. |
| MachineName | Name of a physical or virtual machine. |
| MaintenanceMode | Maintenance mode state. |
| ObjectID | Identifier of an inventory object. |
| ObjectType | Type of an inventory object. |
| PolicyDisplayName | Display name of a policy. |
| PolicyObject | Identifier of a policy object. |
| PolicyValue | Value of a policy object. |
| ProtocolId | Identifier of a display protocol. |
| SecurityServerId | Identifier of a security server. |
| SVIAdminFqdn | FQDN of a QuickPrep domain. |
| SVIAdminID | Identifier of a QuickPrep domain. |
| SVIAdminName | Name of a QuickPrep domain. |
| SVIDeploymentGroupID | Identifier of a View Composer deployment group. |
| SVIOperation | Name of a View Composer operation. |
| SVIParentVM | Parent virtual machine in View Composer. |
| SVIPath | Path of an object in View Composer. |
| SVISnapshot | Snapshot in View Composer. |
| SVIVMID | Identifier of a virtual machine in View Composer. |
| ThinAppDisplayName | Display name of a ThinApp object. |
| ThinAppId | Identifier of a ThinApp object. |
| ThinAppRepositoryName | Name of a ThinApp repository. |
| ThinAppRepositoryPath | Path of a ThinApp repository. |
| Time | Date and time value. |
| UserCount | Maximum number of desktop users over a 24-hour period. |
| UserDiskName | Name of a user data disk. |
| UserDisplayName | User name in the form DOMAIN\username. |
| UserName | Name of a user in Active Directory. |
| VCAddress | URL of a vCenter Server. |

# Sample Database Queries and Views

> **NOTE**  Replace the `dbo.VE_` prefix in the examples with the appropriate prefix for your event database server.

Display all error events from the event_historical table.

```
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'
    );
```

Display all warning events from the event_historical table.

```
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'
    );
```

List all recent events that are associated with the user fred in the domain MYDOM.

```
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'
    );
```

List all recent events where the agent on a machine shut down.

```
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'
    );
```

List all recent events where a desktop failed to launch because the desktop pool was empty.

```
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'
    );
```

List all recent events where an administrator removed a desktop pool.

```
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'
    );
```

List all recent events where an administrator added a ThinApp repository.

```
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'
      );
```

# Using View PowerCLI 3

View PowerCLI provides an easy-to-use PowerShell interface to View. You can use the View PowerCLI cmdlets to perform various administration tasks on View components.

This chapter includes the following topics:

-

-

-

-

-

## Introduction to View PowerCLI

Windows PowerShell is a command-line and scripting environment that is designed for Microsoft Windows. PowerShell uses the .NET object model and provides administrators with management and automation capabilities. As with any other console environment, you work with PowerShell by running commands, which are called cmdlets in PowerShell.

The command-line syntax for the View PowerCLI cmdlets is the same as generic PowerShell syntax. See the Microsoft documentation for more information about using PowerShell.

View includes 45 View PowerCLI cmdlets. You can use these cmdlets in conjunction with the vSphere PowerCLI cmdlets, which provide an administrative interface to the VMware vSphere product. If you have installed vSphere PowerCLI on a Connection Server instance, these cmdlets are loaded automatically when you launch View PowerCLI.

You can reference virtual machines and vCenters by ID in View PowerCLI, but you cannot pass such entries as objects. You must provide a full path to other vSphere objects such as resource pools and folders. You can use the View PowerCLI cmdlets to examine the configuration of vCenter Servers within View.

The View PowerCLI cmdlets are located in the `PowershellServiceCmdlets.dll` file in the directory `C:\Program Files\VMware\VMware View\Server\bin`. This file constitutes the `VMware.View.Broker` snapin. The script configuration file for the View PowerCLI cmdlets is named `InitViewCmdlets.ps1`, and this file is located in the `Extras` folder in the View installation directory. You can edit and extend the script to define cmdlet aliases, to configure the environment, or to set startup actions.

## Start the PowerShell Console with View PowerCLI Loaded

You can run the View PowerCLI cmdlets directly on a View Connections Server host that has the following software installed.

- View 4.5 or later

- Microsoft .NET framework

- Windows PowerShell 1.0

To use the View PowerCLI cmdlets to change the configuration of View, you must be logged into a View Connection Server instance as a user in a role with sufficient privileges, such as the Administrators role. If your role is a read-only role, you cannot update configuration data. For more information, see the *View Administration* document.

### To start the PowerShell console with View PowerCLI loaded

1   Select **Start** > **All Programs** > **VMware** > **View PowerCLI**.

2   If you see an error message stating that the script configuration file cannot be loaded because the execution of scripts is disabled, type the following PowerShell command and restart the PowerShell console.

    **Set-ExecutionPolicy Unrestricted**

### Accessing View PowerCLI cmdlets from a Remote System

If you use the PowerShell remoting feature to access the View PowerCLI cmdlets from a remote system, be sure to take the same precautions for protecting the View PowerCLI operations that you would take for other remoting PowerShell operations.

You can add the View PowerCLI snapin to the Powershell profile on the remote system so that you do not have to load and initialize the View PowerCLI snapin each time you restart the system.

### To add the View PowerCLI snapin to the Powershell profile

1   On the remote system, open the following file:
    C:\Windows\System32\WindowsPowerShells\v1.0\Profile.ps1

2   Add the following line:

    **add-pssnapin vm***

## Displaying Help for a View PowerCLI cmdlet

To list all View PowerCLI cmdlets, type this command in the PowerShell console.

**Get-Command -PSSnapin VMware.View.Broker | more**

You can get help for a specific cmdlet by using the Get-Help cmdlet in the PowerShell console. For example, to get help on the Add-ViewVC cmdlet.

**Get-Help Add-ViewVC | more**

For more detailed information, add the -full parameter.

**Get-Help Add-ViewVC -full | more**

Alternatively, you can use the help alias for Get-Help.

**help Add-ViewVC -full | more**

## Error Handling

View PowerCLI cmdlets handle all errors as non-terminating errors that halt the execution of a cmdlet but do not terminate a pipeline. You can examine the `$error` automatic variable to determine the nature of an error. You can set the standard PowerShell `$ErrorActionPreference` and `$errorView` automatic variables to control how PowerShell handles non-terminating errors and how it displays them in the shell.

## Error When Piping and Specifying Objects at the Same Time

If you attempt to pipe an object into a cmdlet and specify an object of the same type to that cmdlet, the cmdlet fails with the following error.

```
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.
```

For example, the following cmdlet usage would produce this error.

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

## Escaping Characters in vCenter Path Names

If you specify a path to a vCenter folder that includes certain special characters in the name of an entity, you must escape these characters. Table 3-1 shows the escape sequences for special characters.

**Table 3-1.** Escape Sequences for Special Characters

| Special Character | Escape Sequence |
| --- | --- |
| % | %25 |
| / | %2f |
| \ | %5c |

Do not escape the slashes in the path name itself. For example, you would represent the path to the folder `/datacenter_01/vm/img%-12` as `/datacenter_01/vm/img%25-12`.

Table 3-2 lists the cmdlets and parameters that require you to use escape sequences in entity names.

**Table 3-2.** Cmdlet Parameters Requiring Escape Sequences

| Cmdlet | Parameters Requiring Escape Sequences |
| --- | --- |
| Add-AutomaticLinkedClonePool<br>Update-AutomaticLinkedClonePool | -datastoreSpecs<br>-parentVMPath<br>-resourcePoolPath<br>-vmfolderPath |
| Add-AutomaticPool<br>Update-AutomaticPool | -datastorePaths<br>-resourcePoolPath<br>-templatePath<br>-vmfolderPath |
| Send-LinkedCloneRecompose | -parentVMPath |

# View PowerCLI Cmdlets

You can use View PowerCLI cmdlets to administer View on a View Connection Server instance. You can use the `Get-Help` cmdlet to obtain more help about a View PowerCLI cmdlet as described in

lists the available View PowerCLI cmdlets ordered by noun.

**Table 3-3.** View PowerCLI Cmdlets Ordered by Noun

| Cmdlet | Description |
| --- | --- |
| Add-AutomaticLinkedClonePool | Adds an automatically provisioned linked-clone desktop pool. |
| Update-AutomaticLinkedClonePool | Updates an automatically provisioned linked-clone desktop pool. |
| Add-AutomaticPool | Adds an automatically provisioned full virtual machine desktop pool. |
| Update-AutomaticPool | Updates an automatically provisioned full virtual machine desktop pool. |
| Get-ComposerDomain | Returns information about View Composer. |
| Get-ConnectionBroker | Returns information about View Connection Server and security server instances. |
| Update-ConnectionBroker | Updates the configuration of a View Connection Server or security server instance. |
| Get-DesktopPhysicalMachine | Returns a list of physical machines that are available for use with unmanaged desktop pools. |
| Get-DesktopVM | Returns information about virtual machines. |
| Get-EventReport | Returns an event report for a specified view. |
| Get-EventReportList | Returns the views that are available for use with the Get-EventReport cmdlet. |
| Get-GlobalSetting | Returns global configuration information about the View environment. |
| Update-GlobalSetting | Updates global configuration information about the View environment. |
| Get-License | Returns the View licenses on a View Connection Server instance. |
| Set-License | Sets a View license on a View Connection Server instance. |
| Send-LinkedCloneRebalance | Rebalances linked-clone desktops among the available logical drives. |
| Send-LinkedCloneRecompose | Recomposes linked-clone desktops from a snapshot of their parent virtual machine. |
| Send-LinkedCloneRefresh | Refreshes the operating system disks of linked-clone desktops to their original state and size. |
| Add-ManualPool | Adds a manually provisioned pool of managed desktops. |
| Update-ManualPool | Updates a manually provisioned pool of managed desktops. |
| Add-ManualUnmanagedPool | Adds a manually provisioned pool of unmanaged desktops. |
| Update-ManualUnmanagedPool | Updates a manually provisioned pool of unmanaged desktops. |
| Get-Monitor | Returns a list of health monitors for View services. |
| Get-Pool | Returns information about desktop pools. |
| Remove-Pool | Removes a desktop pool. |
| Add-PoolEntitlement | Creates desktop pool entitlements for users. |
| Get-PoolEntitlement | Returns information about the users who are entitled to use desktop pools. |
| Remove-PoolEntitlement | Removes desktop pool entitlement from users. |
| Get-ProfileDisk | Returns information about persistent user data disks. |
| Get-RemoteSession | Returns information about active remote sessions. |
| Send-SessionDisconnect | Disconnects an active remote session. |
| Send-SessionLogoff | Logs out an active remote session. |

**Table 3-3.**  View PowerCLI Cmdlets Ordered by Noun (Continued)

| Cmdlet | Description |
| --- | --- |
| Get–TerminalServer | Returns information about registered RDS hosts. |
| Add–TerminalServerPool | Adds a desktop pool from RDS hosts. |
| Update–TerminalServerPool | Updates a desktop pool that has RDS hosts. |
| Get–User | Returns information about users. |
| Remove–UserOwnership | Removes the ownership of a virtual machine. |
| Update–UserOwnership | Assigns a user (specified as a SID) to a virtual machine. This cmdlet does not support the assignment of users to physical machines. |
| Add–ViewVC | Adds a vCenter Server to View. |
| Get–ViewVC | Returns information about vCenter Servers. |
| Remove–ViewVC | Removes a vCenter Server from View. |
| Update–ViewVC | Updates the configuration of a vCenter Server in View. |
| Send–VMReset | Resets a virtual machine. |
| Export–NetworkLabelSpecFor FullClone | Lists the shared network labels on all the hosts in a specified cluster on which a full clone desktop pool is to be deployed.The output is exported to a configuration file. |
| Export–NetworkLabelSpecFor LinkedClone | Lists the shared network labels on all the hosts in a specified cluster on which a linked clone desktop pool is to be deployed.The output is exported to a configuration file. |

# View Administrator, View PowerCLI Cmdlet, and vdmadmin Operations

Table 3-4 compares the operations that are supported by View Administrator, View PowerCLI cmdlets, and the vdmadmin command on various View objects. For information about View Administrator and the vdmadmin command, see the *View Administration* document.

**Table 3-4.** View Administrator, View PowerCLI Cmdlet, and vdmadmin Operations

| Object | View Administrator Operations | View PowerCLI Cmdlet Operations | vdmadmin Command Operations |
|---|---|---|---|
| Desktop pool | Add | Add | |
| | | | Assign dedicated |
| | Assign ThinApp | | |
| | Disable | Disable | |
| | Enable | Enable | |
| | Entitle user | Entitle user | |
| | Get information | Get information | |
| | Get unentitled policies | | Get unentitled policies |
| | Get unentitled users | | Get unentitled users |
| | Remove | Remove | |
| | | | Remove assignment |
| | Remove entitlement | Remove entitlement | |
| | Restrict entitlement | | |
| | Set policy | Set policy | |
| | Update | Update | |
| | | Assign network label | |
| | | Get network label configuration | |
| Domain filter | | | Get information |
| | | | Remove filter |
| | | | Set filter |
| Events | Get list | Get list | Get list |
| | | Get report | Get report |
| Folder | Add | | |
| | Get information | | |
| | Move | | |
| | Remove | | |
| Kiosk mode | | | Add client account |
| | | | Disable authentication |
| | | | Enable authentication |
| | | | Get information |
| | | | Get defaults |
| | | | Remove client account |
| | | | Set defaults |
| Linked-clone desktop | Rebalance | Rebalance | |
| | Recompose | Recompose | |
| | Recreate | | |
| | Refresh | Refresh | |
| | Restore | | |
| | Set storage overcommit | Set storage overcommit | |
| | | Get network label configuration | |

**Table 3-4.**  View Administrator, View PowerCLI Cmdlet, and vdmadmin Operations (Continued)

| Object | View Administrator Operations | View PowerCLI Cmdlet Operations | vdmadmin Command Operations |
|---|---|---|---|
| Permission | Add<br>Get information<br>Remove | | |
| Persistent user data disk | Attach<br>Delete<br>Detach<br>Get information<br>Replace | Get information | |
| Physical computer (with View Agent) | Get information | Get information | Get information |
| Remote session | Disconnect<br>Get information<br>Log out | Disconnect<br>Get information<br>Log out | |
| Role | Add<br>Modify<br>Remove | | |
| RDS host | Get information | Get information | |
| User | Configure policy<br>Create administrator<br>Get information<br>Remove administrator | Get information | Get information<br><br>Update FSP |
| vCenter Server | Add<br>Get information<br>Remove<br>Update | Add<br>Get information<br>Remove<br>Update | |
| View Agent | | | Create DCT bundle<br>Get copy of log file<br>Get list of log files<br>Get logging level<br>Get status<br>Get version<br>Override IP address<br>Set logging level |
| View Composer domain | | Get information | |
| View Connection Server | Back up configuration<br>Get information<br><br>Restore configuration<br>Update | Get information<br><br>Update | Remove from group |
| View Connection Server group | | | Get GUID of group<br>Set name of group |
| View Global Setting | Get information<br>Update | Get information<br>Update | |
| View service health monitor | Get information | Get information | Get information |

**Table 3-4.** View Administrator, View PowerCLI Cmdlet, and vdmadmin Operations (Continued)

| Object | View Administrator Operations | View PowerCLI Cmdlet Operations | vdmadmin Command Operations |
|---|---|---|---|
| Virtual machine (with View Agent) | Get information<br>Remove ownership<br>Reset<br>Update ownership | Get information<br>Remove ownership<br>Reset<br>Update ownership | Get information |
| VMware Horizon license | Get information<br>Set license | Get information<br>Set license | |

# View PowerCLI Cmdlet Parameters

Some View PowerCLI cmdlet parameters accept arguments that are chosen from a limited set.

## Default Display Protocol Settings

Table 3-5 shows the settings that you can use with the –defaultProtocol parameter to specify the default display protocol for a desktop pool.

**Table 3-5.** Default Display Protocol Settings

| Default Display Protocol | Description |
|---|---|
| PCOIP | Specifies VMware PCoIP. |
| RDP | Specifies Microsoft RDP. |

## Deletion Policy Settings

Table 3-6 shows the settings that you can use with the –deletePolicy parameter to specify a deletion policy for automatically provisioned floating and linked-clone desktop pools.

**Table 3-6.** Deletion Policy Settings

| Deletion Policy | Description |
|---|---|
| Default | Specifies that the machine is not deleted when the user logs out. |
| DeleteOnUse | Specifies that the machine is deleted when the user logs out. |
| RefreshOnUse | Specifies that the machine is refreshed when the user logs out. This setting applies only to linked-clone desktop pools. |

## Flash Quality Settings

Table 3-7 shows the settings that you can use with the –flashQuality parameter to specify a maximum allowable quality for Adobe Flash content. This value overrides the setting on a Web page. If the Flash quality for a Web page is higher than the maximum value allowed, the client reduces the quality to the specified maximum. Lowering the quality of Flash content causes it to use less bandwidth.

**Table 3-7.** Flash Quality Settings

| Flash Quality | Description |
|---|---|
| HIGH | Specifies that low, medium, or high quality Flash content is allowed. |
| LOW | Specifies that only low quality Flash content is allowed. |
| MEDIUM | Specifies that low or medium quality Flash content is allowed. |
| NO_CONTROL | Allows the Web page settings to determine quality. |

## Flash Throttling Settings

Table 3-8 shows the settings that you can use with the –flashThrottling parameter to specify how often Adobe Flash should refresh what it shows onscreen. Throttling Flash to increase the refresh interval reduces the frame rate. This reduction causes Flash content to use less bandwidth, but it might also cause Flash to drop frames.

**Table 3-8.** Flash Throttling Settings

| Flash Throttling | Description |
| --- | --- |
| AGGRESSIVE | Sets the interval to 2500 milliseconds. This results in the highest number of dropped frames. The speed of audio transmission is unaffected. |
| CONSERVATIVE | Sets the interval to 100 milliseconds. This results in the lowest number of dropped frames. The speed of audio transmission is unaffected. |
| DISABLED | No throttling is performed. The timer interval is not modified. |
| MODERATE | Sets the interval to 500 milliseconds. The speed of audio transmission is unaffected. |

## LDAP Backup Frequency

Table 3-9 shows the settings that you can use with the –ldapBackupFrequency parameter to specify the LDAP backup frequency for a View Connection Server instance.

**Table 3-9.** LDAP Backup Frequency Settings

| LDAP Backup Frequency | Description |
| --- | --- |
| Every12Hour | Backs up the LDAP database once every 12 hours. |
| Every2Day | Backs up the LDAP database once every two days. |
| Every2Week | Backs up the LDAP database once every two weeks. |
| Every6Hour | Backs up the LDAP database once every six hours. |
| EveryDay | Backs up the LDAP database once per day. |
| EveryHour | Backs up the LDAP database once per hour. |
| EveryWeek | Backs up the LDAP database once per week. |
| Never | Turns off backup for the LDAP database. |

## Pool Type Settings

Table 3-10 shows the settings that you can use with the –poolType parameter to specify the type of a desktop pool.

**Table 3-10.** Pool Type Settings

| Pool Type | Description |
| --- | --- |
| IndividualUnmanaged | Specifies a pool containing an individual unmanaged machine. |
| IndividualVC | Specifies a pool containing an individual machine that is managed and configured by a vCenter Server. |
| Manual | Specifies a pool of manually configured floating (nonpersistent) machines that are managed and configured by a vCenter Server. |
| ManualUnmanagedNonPersistent | Specifies a pool of manually configured floating (nonpersistent) machines that are not managed. |
| ManualUnmanagedPersistent | Specifies a pool of manually configured dedicated (persistent) machines that are not managed. |
| ManualVCPersistent | Specifies a pool of manually configured dedicated (persistent) machines that are managed by a vCenter Server. |

**Table 3-10.** Pool Type Settings (Continued)

| Pool Type | Description |
|---|---|
| NonPersistent | (AutomaticPool) Specifies a pool of automatically configured floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server. |
| OnRequestSviNonPersistent | (AutomaticPool) Specifies a pool of floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server and View Composer when requested. |
| OnRequestSviPersistent | (AutomaticPool) Specifies a pool of dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server and View Composer when requested. |
| OnRequestVcNonPersistent | (AutomaticPool) Specifies a pool of floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server when requested. |
| OnRequestVcPersistent | (AutomaticPool) Specifies a pool of dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server when requested. |
| Persistent | (AutomaticPool) Specifies a pool of automatically configured dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server. |
| SVINonPersistent | (AutomaticPool) Specifies a pool of floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server and View Composer. |
| SVIPersistent | (AutomaticPool) Specifies a pool of dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server and View Composer. |
| TerminalService | Specifies a pool of machines that are managed and configured by an RDS host. |

## Power Policy Settings

Table 3-11 shows the settings that you can use with the —powerPolicy parameter to specify the power policy for a desktop pool.

**Table 3-11.** Power Policy

| Power Policy Setting | Description |
|---|---|
| AlwaysOn | Configures a machine to remain powered on, even when no one is using it. If you shut down the machine, it restarts immediately. |
| RemainOn | Starts a machine when required if it is powered down. The machine then remains powered on until you shut it down. |
| PowerOff | Shuts down a machine when no one is using it. |
| Suspend | Suspends a machine when no one is using it. |

## Refresh Policy Settings

Table 3-13 shows the settings that you can use with the —refreshPolicyType parameter to specify a refresh policy for the OS disks of automatically provisioned dedicated and linked-clone desktop pools.

**Table 3-12.** Refresh Policy Settings

| Refresh Policy | Description |
|---|---|
| Always | Specifies that an OS disk is always refreshed when the user logs out. |
| Conditional | Specifies that an OS disk is refreshed when the user logs out if certain conditions are met. Use the —refreshpolicyDays and —refreshPolicyUsage parameters to specify the refresh interval in days and the percentage of the maximum allowable size for the disk. |
| Never | Specifies that an OS disk is never refreshed when the user logs out. |

## Smart Card Authentication Settings

Table 3-13 shows the settings that you can use with the `–smartCardSetting` parameter to specify how users with smart cards are authenticated. These settings apply to `Update-ConnectionBroker,` and do not have any effect unless you also specify `–UseSSLClient $true` with `Update-GlobalSetting`.

**Table 3-13.** Smart Card Authentication Settings

| Refresh Policy | Description |
| --- | --- |
| NotAllowed | Specifies that smart card authentication is disabled. |
| Optional | Specifies that users can use smart card authentication or password authentication to connect to the View Connection Server instance. If smart card authentication fails, the user must provide a password. |
| Required | Specifies that users must use smart card authentication when connecting to the View Connection Server instance. Smart card authentication replaces Windows password authentication only. If SecureID is enabled, users are required to authenticate by using both SecureID and smart card authentication. |

## View Composer Maintenance Task Types

Table 3-14 shows the settings that you can use with the `–composerTask` parameter to specify a View Composer maintenance task type that is scheduled on a virtual machine.

**Table 3-14.** View Composer Maintenance Task Types

| Task Type | Description |
| --- | --- |
| attachUdd | Specifies a scheduled task to attach a persistent disk. |
| detachUdd | Specifies a scheduled task to detach a persistent disk. |
| mkChkPoint | Specifies a scheduled task to create a checkpoint snapshot. |
| rebalance | Specifies a scheduled task to rebalance a linked-clone machine. |
| refresh | Specifies a scheduled task to refresh a linked-clone machine. |
| replaceUdd | Specifies a scheduled task to replace a persistent disk. |
| resync | Specifies a scheduled task to recompose a linked-clone machine. |

# Examples of Using View PowerCLI Cmdlets

The View PowerCLI cmdlets allow you to perform View operations from the command line or from scripts instead of using View Administrator. The following sections contain examples that you can adapt and apply to your own systems.

## Displaying Information About a View Connection Server Instance

Display the configuration settings of a View Connection Server instance.

```
Get-ConnectionBroker –broker_id CONNSVR1
```

## Updating the Configuration of a View Connection Server Instance

Update the settings for direct connections and secureID on a View Connection Server instance.

```
Update-ConnectionBroker –broker_id CONNSVR1 –directConnect $false –secureIdEnabled $true
–ldapBackupFrequency EveryWeek
```

Configure a View Connection Server for secure PCoIP connections.

```
Update-ConnectionBroker –broker_id CS-VSG –directPCoIP $FALSE
```

Set the PCoIP external URL for a View Connection Server.

```
Update-ConnectionBroker –broker_id CS-VSG –externalPCoIPURL 10.18.133.34:4172
```

Set the PCoIP external URL for a security server.

```
Update-ConnectionBroker -broker_id SECSVR-03 -externalPCoIPURL 10.116.32.136:4172
```

## Managing the Configuration of vCenter Servers in View

Add an entry for a vCenter Server to the View configuration.

```
Add-ViewVC -serverName vc01.mydom.int -username Administrator -password clydenw
-createRampFactor 5 -deleteRampFactor 5
```

Return information about a vCenter Server.

```
Get-ViewVC -serverName vc01.mydom.int
```

Return information about all vCenter Servers in a specified DNS domain.

```
Get-ViewVC -serverName *.mycorp.com
```

Change the values of the ramp factors for the vCenter Server that is configured on server svr11.

```
Get-ViewVC -serverName svr11.mycorp.com | Update-ViewVC -createRampFactor 5 -deleteRampFactor 10
```

Change the value of the create ramp factor for the vCenter Servers that are configured in the DNS domain mycorp.com.

```
Get-ViewVC -serverName *.mycorp.com | Update-ViewVC -createRampFactor 5
```

Remove an entry for a vCenter Server from the configuration.

```
Get-ViewVC -serverName vc02.mydom.int | Remove-ViewVC
```

## Managing Desktop Pools

Return information about the desktop pool with a specified display name.

```
Get-Pool -displayName "My Pool 1"
```

Return information about all desktop pools with the prefix mypool-.

```
Get-Pool -pool_id mypool-*
```

Return information about all desktop pools that are configured to use the PCoIP protocol.

```
Get-Pool -protocol PCOIP
```

Return information about all individual unmanaged desktop pools.

```
Get-Pool -poolType IndividualUnmanaged
```

Remove the desktop pool dtpool-10.

```
Remove-Pool -pool_id dtpool-10
```

Remove the desktop pool dtpool-12, terminating any active session, but without deleting the image from disk.

```
Remove-Pool -pool_id dtpool-12 -TerminateSession $true -DeleteFromDisk $false
```

Remove a desktop pool specified by its display name, and delete its image from disk.

```
Get-Pool -displayName "My Pool 1" | Remove-Pool -DeleteFromDisk $true
```

## Creating and Updating Automatically Provisioned Desktop Pools

Add the automatically provisioned desktop pool auto1, which is managed by the vCenter Server vc.mydom.int.

```
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
```

**NOTE**   To provision all the desktops in advance of their use, set –`minimumCount` and –`maximumCount` to the same value.

Update the configuration of the automatically provisioned desktop pool auto1.

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

The datastores that you specify as arguments to –`datastorePaths` override the previous setting. You must specify the existing datastores to retain them for use by the pool.

## Creating and Updating Linked-Clone Desktop Pools

Add the linked-clone desktop pool lcdpool_1, which is managed by View Composer on the vCenter Server vc.mydom.int.

```
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
```

**NOTE**   To provision all the desktops in advance of their use, set –`minimumCount` and –`maximumCount` to the same value.

If you specify a persistent data disk, use an uppercase letter for the drive. Do not use a letter that already exists on the parent virtual machine for a drive such as A, B, or C, or that conflicts with a network-mounted drive.

Update the configuration of the linked-clone desktop pool lcdpool_1.

```
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
```

The datastores that you specify as arguments to –`datastoreSpecs` override the previous setting. You must specify the existing datastores to retain them for use by the pool.

Rebalance desktops in the linked-clone desktop pool lcdpool_2 among the available datastores.

```
Get–DesktopVM –pool_id lcdpool_2 | Send–LinkedCloneRebalance –schedule 2011–05–10:01:00:00
–forceLogoff $false –stopOnError $true
```

Refresh the operating system disk of each linked-clone desktop by restoring it to its original state and size.

```
Get–DesktopVM –pool_id lcdpool_2 | Send–LinkedCloneRefresh –schedule "May 12 2011 01:15"
–forceLogoff $true –stopOnError $true
```

Recompose all linked-clone desktops from a snapshot of their parent virtual machine.

```
Get–DesktopVM –pool_id lcdpool_2 | Send–LinkedCloneRecompose –schedule ((Get–Date).AddHours(8))
–parentVMPath /AutoPoolVMs/parent2 –parentSnapshotPath /AutoPoolSnapshots/parent2_snapshot
–forceLogoff $true –stopOnError $true
```

## Creating and Updating Manually Provisioned Desktop Pools

Create a manually provisioned desktop pool that contains the virtual machine myVM. To use the `Get–VM` cmdlet, you must install vSphere PowerCLI.

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

Create the manually provisioned desktop pool man1 from the desktops listed by the vCenter Server vc.mydom.int.

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

Update the configuration of the manually provisioned desktop pool man1.

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

## Creating Manual Unmanaged Desktop Pools

Create the unmanaged desktop pool unman1 that contains the physical machines pm01 and pm02.

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

Update the configuration of the unmanaged desktop pool unman1.

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

## Creating and Updating Desktops Provisioned by RDS Hosts

Return information about an RDS host (formerly called a terminal server).

```
Get-TerminalServer -hostname tsvr01
```

Add the desktop pool tsd1 using a machine that is provided by the RDS host at a specified IP address.

```
Get-TerminalServer -hostname tsvr01 | Add-TerminalServerPool -pool_id tsd1 -displayName "Terminal
Server Desktop 1" -allowProtocolOverride $true
```

Update the configuration of the desktop pool tsd1, which uses a machine provided by an RDS host.

```
Get-TerminalServer -hostname tsvr01 | Update-TerminalServerPool -pool_id tsd1 -displayName
"Terminal Server Desktop 1" -allowProtocolOverride $false -autoLogoffTime 1
```

## Obtaining Information About Users and Groups from Active Directory

Return information about users in the domain mydom.

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

Return information about the user fred in the domain mydom and exclude any information about the user's group.

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

## Managing Desktop Entitlements

Add an entitlement for the user fred in the domain mydom to use desktop pool dtop-12.

```
Get-User -name "mydom\fred" | Add-PoolEntitlement -pool_id dtop-12
```

Add an entitlement to all desktop pools for the user usr1.

```
Get-Pool | Add-PoolEntitlement -sid (Get-User -name "usr1").sid
```

Return information about all entitlements to desktop pools.

```
Get-PoolEntitlement
```

Return information about the users who are entitled to use desktop pool dtop-1.

```
Get-PoolEntitlement -pool_id dtop-1
```

Return information about the users who are entitled to use desktop pools whose IDs start with dtpool-.

```
Get-Pool -pool_id dtpool-* | Get-PoolEntitlement
```

Remove the entitlement to use desktop pool dtpool-11.

```
Get-PoolEntitlement -pool_id dtpool-11 | Remove-PoolEntitlement
```

If you want to remove all entitlements from the system, specify the –forceRemove parameter set to $true .

```
Get-PoolEntitlement | Remove-PoolEntitlement -forceRemove $true
```

If you do not specify the–forceRemove parameter, you can use the command to find out information about the entitlements that would be removed.

## Managing Remote Sessions

Return the list of active remote sessions for the user fred in the domain mydom.

```
Get-RemoteSession -username mydom\fred
```

Disconnect the active remote sessions on which the user fred is logged in.

```
Get-RemoteSession -username mydom\fred | Send-SessionDisconnect
```

Log out the active remote sessions on which the user fred is logged in.

```
Get-RemoteSession -username mydom\fred | Send-SessionLogoff
```

Log out the active remote sessions that are using the RDP protocol.

```
Get-RemoteSession -protocol RDP | Send-SessionLogoff
```

## Managing Virtual Machines

---
NOTE   View Agent must be running on the virtual machines.

---

Return information about the virtual machine for the desktop pool dtpool-3.

```
Get-DesktopVM -pool_id dtpool-3
```

Return information about the virtual machines that are configured on the vCenter Server vc03.local.int.

```
Get-DesktopVM -vc_id (Get-ViewVC -serverName vc03.local.int).vc_id
```

Return information about the virtual machines that are managed by the same vCenter Server that provisions the desktop pool dtpool-1.

```
Get-ViewVC -pool_id dtpool-1 | Get-DesktopVM
```

Return information about all virtual machines that are managed by the vCenter Server that is configured on the server vc01.mydom.int.

```
Get-ViewVC -serverName vc01.mydom.int | Get-DesktopVM
```

Return a list of active persistent user data disks for the virtual machine vm01.

```
Get-ProfileDisk -VMname vm01
```

Reset the virtual machines for the desktop pool dtpool-05.

```
Get-Pool -pool_id dtpool-05 | Get-DesktopVM | Send-VMReset
```

Reset the virtual machine for the desktop pool with the display name dtp1.

```
Get-Pool -displayName dtp1 | Get-DesktopVM | Send-VMReset
```

## Displaying Information About Physical Computers

---
NOTE   View Agent must be running on the computers.

---

Return information about the computer at a specified IP address.

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

## Updating the Ownership of Machines

Update the ownership of the virtual machine vm04 to the user usr1.

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

Remove the ownership of a virtual machine.

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

## Displaying Information About Events

Display the views that are available for reporting events.

```
Get-EventReportList
```

Display configuration change events that have occurred since the specified date.

```
Get-EventReport -viewName config_changes -startDate (Get-Date -Year 2011 -Month 5 -Day 20 -Hour 0
-Minute 0 -Second 0)
```

Display user events that occurred between two specified dates.

```
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)
```

Display user events for the last 24 hours.

```
Get-EventReport -viewName user_events -startDate ((Get-Date).AddDays(-1))
```

Display user events for the current year.

```
Get-EventReport -viewName user_events -startDate (Get-Date -Day 01 -Month 01 -Hour 0 -Minute 0
-Second 0)
```

## Managing the Global Configuration of View

Display the global configuration settings.

```
Get-GlobalSetting
```

Set the session timeout to 30 minutes.

```
Update-GlobalSetting -SessionTimeout 1800
```

NOTE   The -SessionTimeout value is measured in seconds. In View Administrator, the Session timeout value is measured in minutes. If View PowerCLI generates a value that is less than 60 seconds, View Administrator rounds down, resulting in a value of 0. To set a session timeout in View PowerCLI, specify a -SessionTimeout value of 60 (one minute) or greater.

Set the forced logout warning message and delay period.

```
Update-GlobalSetting -DisplayLogoffWarning $true -ForcedLogoffAfter $logoutdelay
-ForcedLogoffMessage "Forced log out will occur in $logoutdelay minutes"
```

Require clients to use SSL to connect and set the prelogin message.

```
Update-GlobalSetting -UseSSLClient $true -PreLoginMessage "Insert disclaimer and other notices
here."
```

## Managing View Licenses

Display the installed View license keys.

```
Get-License
```

Add a license key.

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

# Examples of Using View PowerCLI for Enhanced Functionality

You can create PowerShell functions by combining View PowerCLI and vSphere PowerCLI cmdlets to perform complex operations such as resizing pools, and adding datastores to desktop pools. The following sections contain sample functions that you can adapt and apply to your own systems.

## Checking if a View Connection Server Instance Is Running

Define a PowerShell function to check if a View Connection Server instance is running, and optionally, start the service.

```
# 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 = $error.Count
                if($errorCountAfter -gt $errorCountBefore){
                    break
                }
            } else {
                Write-Error "Connection Broker service is stopped."
                break
            }
        }
        while(!$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?"
    }
}
```

## Resize Automatic and Linked-Clone Desktop Pools

Define PowerShell functions to check the current usage of all desktop pools, and to resize any automatically provisioned or linked-clone desktop pools that are at their maximum capacity.

```
# 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.")
        }
```

```
        }
}
```

## Determining Paths to vSphere Inventory Objects

Define a PowerShell function that uses vSphere PowerCLI to return the full path to a vSphere inventory object. For a function that you can use to determine datastore paths, see

```
# 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
    }
}
```

## Determining Paths to vSphere Datastore Objects

Define a PowerShell function that uses vSphere PowerCLI to return the full path to a datastore in a cluster as specified by a resource pool.

```
# 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
    }
}
```

## Adding and Removing Datastores

Define a PowerShell function to add a datastore to an automatic pool.

```
# 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
}
```

Define a PowerShell function to add a datastore to a linked-clone pool.

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

## Assign Multiple Network Labels to a Desktop Pool

In View 5.2 and later releases, you can configure automated desktop pools to use multiple network labels. This feature greatly expands the number of IP addresses that can be assigned to the virtual machines in a pool, making it easier to create pools with a large number of desktops.

By default, the virtual machines in a desktop pool inherit the network interface card (NIC), and its associated network label, that are on the parent virtual machine or template. (Some parent virtual machines or templates might have multiple NICs with their associated network labels.) Typically, the subnet mask of a VLAN defined by a network label has a limited range of available IP addresses. For example, a subnet mask might have a maximum of 254 IP addresses that can be assigned to the desktop virtual machines.

In View 5.2 and later releases, View PowerCLI cmdlets let you assign network labels that are available in the vCenter Server resource pool where the desktop pool is being deployed. Specifically, you take the following steps:

- Select network labels from those that are defined for the ESXi cluster in vCenter Server

- Associate the labels with the NICs that are inherited from the parent virtual machine or template

- Specify a maximum number of IP addresses that can be assigned to virtual machines from each network label

- Save the preceding information in a network label configuration file

- Use the configuration file with a View PowerCLI cmdlet that creates a desktop pool

You can assign multiple network labels to automated full clone pools or linked clone pools.

The network labels are distributed among the virtual machines in the entire desktop pool. When View provisions desktops, network labels are assigned in alphabetical order. When the maximum number of virtual machines are provisioned with IP addresses using the first network label, View starts assigning the second label, and so on.

To assign and manage multiple network labels in a pool, perform the following tasks:

- "Obtain NIC and Network Label Information in a Configuration File" on page 52

- "Edit the Network Label Configuration File" on page 54

- "Deploy a Desktop Pool with Network Label Assignments" on page 55

- "Display the Network Label Assignments for an Existing Pool or Desktop" on page 56

■

## Obtain NIC and Network Label Information in a Configuration File

Before you can assign multiple network labels to a pool, you must obtain information about the NICs configured on the template or parent virtual machine and the available network labels configured on the ESXi hosts in the cluster on which the desktop pool will be deployed. This information is saved in a configuration file that you use when you create the pool.

Use the `Export-NetworkLabelSpecForLinkedClone` cmdlet to obtain the NICs on a parent virtual machine and snapshot used to create a linked-clone pool.

Use the `Export-NetworkLabelSpecForFullClone` cmdlet to obtain the NICs on a template used to create a full clone pool.

These commands also perform the following operations:

■ Obtain the network labels configured in vCenter Server for all the ESXi hosts in a specified cluster

■ Set a specified maximum number of virtual machines that can be provisioned with IP addresses from each network label

■ Export the list of NICs and network labels to a specified configuration file

### Example of Obtaining NIC and Network Label Information

The following example obtains NICs from the `Win7-Parent` virtual machine and `Snapshot1`, and network label information from `Cluster1`. A maximum of 239 virtual machines can be provisioned with IP addresses from each network label. The information is exported to the configuration file `C:/label.txt`.

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

When you set the `-maxVMsPerNetworkLabel` parameter, allow a certain number of virtual machines for overhead. For example, if a network label VLAN allows a maximum of 254 IP addresses to be assigned to virtual machines, you might set the maximum to 244.

The `Export-NetworkLabelSpecForFullClone` cmdlet is similar to the cmdlet shown in the preceding example, but it uses the `-TemplatePath` parameter to specify the template from which you will generate a full clone pool.

For details about the cmdlet syntax and parameters, display the View PowerCLI help.

### Contents of a Network Label Configuration File

The configuration file generated by the Export Network Label cmdlets contains the following sections. For a sample configuration file, see "Example Network Label Configuration File" on page 53.

■ **Enabled flag**. This flag is set to `enabled=true` by default. Keep the flag set to `true` to allow View to assign network labels to the pool.

■ **Parameter Definition for NIC**. Lists the NICs defined in vCenter Server on the template or the snapshot of the parent virtual machine. Do not edit this section.

■ **Parameter Definition for Network**. Lists the network labels defined in vCenter Server for the ESXi hosts in the cluster. Network labels are listed in alphabetical order. If a cluster uses standard network labels and distributed virtual switch network labels, use only one type of label for a pool. Do not edit this section.

■ **Network Label Attribute Definition**. Lists the network labels associated with each NIC. The network labels are commented out and the assignments are not functional. You must remove the comments (# marks) from the appropriate network labels to allow them to be assigned to a desktop pool.

The `maxvm` parameter defines the maximum number of IP assignments that can be made to virtual

machines from the network label. The value of the `maxvm` parameter is generated by the `–maxVMsPerNetworkLabel` parameter in the Export Network Label cmdlet. You can manually edit this value in the configuration file.

If the parent virtual machine or template has two NICs, each network label discovered by the Export Network Label cmdlet is associated with both NICs. Warning messages explain that IP address assignments can become oversubscribed because the assignment function for one NIC is not aware of the assignments that are made from the same network label for the second NIC. Each NIC is only aware of its own network label assignments.

**Example Network Label Configuration File**

```
#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
```

## Edit the Network Label Configuration File

To edit the network label configuration file

1  Open the file in a text editor and remove the comments (### marks) for each NIC and network label pairing that you want to assign to a pool.

2  Verify that each network label is assigned to only one NIC. Verify that only one type of network label is used for the pool. (A NIC can be configured with a standard network switch or distributed virtual network switch.)

3  Save the file.

For an example, see "Example of an Edited Network Label Configuration File" on page 54.

**NOTE**  As a best practice, do not assign a network label to more than one desktop pool. Be aware that maximum network label counts are honored only on a per-NIC, per-pool basis. For example, if you configure NIC1 on pool1 to use network06 with a maxvm of 244, and you configure NIC1 on pool2 to use the same network label, network06, with a maxvm of 244, network06 must have an actual assignable IP address space of at least 488 addresses. Otherwise, the IP assignments from the network label might become oversubscribed.

### Example of an Edited Network Label Configuration File

In the following example, network01 through network05 remain commented and are not used. network06 and network07 are assigned to nic1 (Network adapter 1). network08 and network09 are assigned to nic2 (Network adapter 2). This configuration can now support a pool of up to 488 virtual machines.

```
#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
```

## Deploy a Desktop Pool with Network Label Assignments

To create an automated desktop pool that can use multiple network labels, use the `Add-AutomaticLinkedClonePool` cmdlet for linked clone pools or the `Add-AutomaticPool` cmdlet for full clone pools.

Use the `-NetworkLabelConfigFile` parameter with the Add Pool cmdlet to assign the network labels in a specified configuration file to the pool. Specify the path to the configuration file with the "*path*"argument: `-NetworkLabelConfigFile` "*path*".

The `-NetworkLabelConfigFile` parameter works the same way for linked clone and full clone pools.

The following example creates a linked clone pool, `POOL_NAME`, and assigns the network labels in the `label.txt` file to the pool.

```
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;[Mod
          erate,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"
```

### Operation of Network Label Assignments

As View provisions virtual machines in a pool, network labels are assigned to the NICs in the virtual machines. When the maximum number of virtual machines to be assigned a certain network label is reached, View begins provisioning from the next network label.

After a network label is assigned to a virtual machine, View does not change the assignment.

If virtual machines in a pool are deleted, the associated network label assignments are freed and can be assigned to any newly provisioned virtual machines in the pool.

View Composer refresh operations do not affect network label assignments. Virtual machines continue to receive the network labels that were assigned to them when they were first provisioned.

### Network Label Preservation During Recompose and Rebalance Operations

During View Composer recompose and rebalance operations, a best effort is made to ensure that the network label of each NIC attached to each linked-clone desktop is preserved when a linked clone inherits new NICs from a new base image. View preserves the network label of a NIC that was in place before the recompose or rebalance operation as long as the new base image has an available NIC configured with the same type of network switch. (A NIC can be configured with a standard network switch or distributed virtual network switch.)

The following examples illustrate the rules governing network label preservation during recompose and rebalance operations:

- If both the original linked-clone desktop and the new base image have one NIC configured with a standard network switch, the network label and MAC address are preserved.

- If both the original linked-clone desktop and the new base image have one NIC configured with a distributed virtual network switch, the network label and MAC address are preserved.

- If the original linked-clone desktop has one NIC configured with a standard network switch and the new base image has one NIC configured with a distributed virtual network switch, the network label is not preserved. The MAC address is preserved.

- If the original linked-clone desktop has two NICs, both of which are configured with a standard network switch, and the new base image has one NIC configured with a standard network switch, the network label of one original NIC is preserved, but all network properties associated with the other original NIC are not preserved.

## Display the Network Label Assignments for an Existing Pool or Desktop

To display the network label assignments for a pool, use the `Get-Pool` cmdlet. The `NetworkLabelSpecs` output parameter shows the network labels that are assigned to the pool. You do not have to add a parameter in the command line to display this information. It is displayed automatically with the `Get-Pool` cmdlet.

The `NetworkLabelSpecs` output displays infomation in the following format:

- `nl` – Network label name.

- `nic` – NIC name

- `enabled` – A value of 1 means the label is active. A value of 0 means it is disabled.

- `max` – Maximum number of virtual machines that can be assigned the network label

- `usage` – Number of virtual machines currently assigned to the network label.

  When the `usage` value equals the `max` value, View stops assigning that network label and starts making assignments from the next available label.

The following example displays network label assignments for the pool `Pool2`.

```
> 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=desktop-
                            auto05-270;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] ...
```

### Displaying Network Label Information for a Virtual Machine

To display the network label assignments for a virtual machine, use the `Get-DesktopVM` cmdlet. The `netLabelAutoAssigns` output parameter shows the NIC names and network labels that View has attempted to assign to the virtual machine.

You do not have to add a parameter in the command line to display this information. It is displayed automatically with the `Get-DesktopVM` cmdlet.

If the `Enabled` flag in the network label configuration file was set to 0, the `netLabelAutoAssigns` output shows this feature is disabled.

NOTE  The `netLabelAutoAssigns` output displays the network label assignments that View Connection Server reserves for the virtual machine. To see the network labels that are actually assigned to the virtual machine in vCenter Server, use the `–getNetworkLabel` parameter with the `Get–DesktopVM` cmdlet. See "Displaying Network Label Assignments Made for a Virtual Machine in vCenter Server" on page 57.

The following example displays network label assignments for the virtual machine 918 in the pool `pool2`.

```
> 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} ...
```

### Displaying Network Label Assignments Made for a Virtual Machine in vCenter Server

The `netLabelAutoAssigns` output displays the network label assignments that View Connection Server reserves for the virtual machine. Additional network label assignments to other virtual machines can be made in vCenter Server that are outside the control of View.

To see the network labels that are actually assigned to the virtual machine in vCenter Server, use the `–getNetworkLabel` parameter with the `Get–DesktopVM` cmdlet.

In the command line, you must type a Boolean value of `$true` to enable the `–getNetworkLabel` parameter. For example: `–getNetworkLabel $true`

The output of the `Get–DesktopVM` cmdlet displays the `networkLabels` parameter, which shows the NICs and network label assignments that were made for the virtual machine.

NOTE  As a best practice, run the `Get–DesktopVM` cmdlet with the `–getNetworkLabel` parameter during off-peak periods of vSphere utilization. The parameter is long-running.

The following example displays network label assignments made in vCenter Server for the virtual machine 1849 in the pool `pool2`.

```
> 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            : POOL2-85.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} ...
```

## Disable Automatic Network Label Assigments

To disable network label assignments on an existing pool that uses automatic assigments

1   Open the network label configuration file and set the `enabled` flag to `enabled=false`.

2   Save the file.

3   Run the `Update-AutomaticLinkedClonePool` or `Update-AutomaticPool` cmdlet with the
    `-NetworkLabelConfigFile` parameter pointing to the updated configuration file.

**Result:**

When View provisions new virtual machines in the pool, View uses the network labels on the parent virtual
machine or template.

# Customizing LDAP Data 4

You can use VMware and Microsoft command tools to export and import LDAP configuration data in LDAP Data Interchange Format (LDIF) files from and into View. These commands are intended for use by advanced administrators who want to use scripts to update configuration data without using View Administrator. If you want to create scripts to update the View configuration, VMware recommends that you use View PowerCLI rather than LDAP commands.

This chapter includes the following topics:

## LDAP Configuration Data

All View configuration data is stored in an LDAP directory. Each View Connection Server standard or replica instance contains a local LDAP configuration repository and a replication agreement between each of the View Connection Server instances. This arrangement ensures that changes to one repository are automatically replicated to all the other repositories.

When you use View Administrator to modify the configuration of View, the appropriate LDAP data in the repository is updated. For example, if you add a desktop pool, View stores information about users, user groups, and entitlements in LDAP. The View Connection Server instances manage other LDAP configuration data automatically, and they use the information in the repository to control View operations.

You can use LDIF files to perform a number of tasks.

- To transfer configuration data between View Connection Server instances.

- To define a large number of View objects, such as desktop pools, and add these to your View Connection Server instances without using View Administrator to do this manually.

- To back up your View configuration so that you can restore the state of a View Connection Server instance.

In View 3.1 and later, regular backups of the LDAP repository are made automatically.

LDAP configuration data is transferred as plain ASCII text and conforms to the Internet Engineering Task Force (IETF) RFC 2849 standard.

# Export LDAP Configuration Data

You can export configuration data from a standard or replica View Connection Server instance to an LDIF file by running the `vdmexport` command-line utility.

By default, the `vdmexport` command-line utility is installed in the `C:\Program Files\VMware\VMware View\Server\tools\bin` directory.

In View 3.1 and earlier, you must be logged into a standard or replica View Connection Server instance as an administrator and be a member of the Local Administrators user group.

In View 4.5 and later, you must be logged in to a View Connection Server instance as a user in the Administrators or Administrators (Read only) role to export configuration data successfully from the View configuration repository.

**To export configuration data from a View Connection Server instance**

1    Select **Start > Command Prompt**.

2    At the command prompt, type the `vdmexport` command and redirect the output to a file.

    vdmexport > myexport.LDF

     You can specify the output file name as an argument to the `−f` option.

    vdmexport −f myexport.LDF

     The command writes the configuration of your View Connection Server instance to the file that you specify. The command displays errors if your role has insufficient privileges to view the data in the configuration repository.

# Modify LDAP Configuration Data

You can modify the LDAP configuration data that you have exported from a View Connection Server instance to an LDIF file and use the modified file to perform automatic bulk configuration operations on View Connection Server instances.

You can obtain an example of the LDIF syntax for any item of LDAP configuration data in View. For example, you can extract the data for a desktop pool and use this as a template to create a large number of desktop pools.

**To modify LDAP configuration data**

1    Use View Administrator to add a single desktop pool with the default attribute values that you require.

     If required, you can modify these values when you create your customized configuration file.

2    Use the `vdmexport` command to export the configuration data as an LDIF file.

     See "Export LDAP Configuration Data" on page 60.

3    Examine the contents of the LDIF file to find the entry definitions of the desktop pool that you created.

     You can use this entry as a template for adding a large number of desktop pools.

Use the `vdmimport` command to import your customized LDIF file and update the configuration of the View Connection Server instance. See "Import LDAP Configuration Data" on page 63.

You must add the following entries to an LDIF file to define a desktop pool:

■    A Virtual Desktop VM entry for each virtual desktop in the desktop pool.

■    A VM Pool entry for each desktop pool.

■    A Desktop Application entry that defines the entitlement of the desktop pool.

Each VM Pool entry must be associated with one Desktop Application entry in a one-to-one relationship. A Desktop Application entry cannot be shared between VM Pool entries, and a VM Pool entry can only be associated with one Desktop Application entry.

The following extract from an LDIF file shows sample entries for a desktop pool named Pool1, which contains two virtual desktops named VM1 and VM2. The desktop pool entry is paired with the Desktop Application entry, which is also named Pool1.

```
#
# 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: KERB5
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
```

Table 4-1 lists the attributes that are important when you modify a desktop pool definition.

**Table 4-1.** Important Attributes for Defining a Desktop Pool

| Entry | Attribute | Description |
|---|---|---|
| Virtual Desktop VM<br>VM Pool<br>Desktop Application | cn | Specifies the common name of an entry. If you require names to be generated automatically, specify globally unique identifier (GUID) strings. You can use any reliable GUID generator, such as the mechanism provided by .NET (for example, by calling `System.Guid.NewGuid().ToString()` in Visual Basic). |
| Desktop Application | member | Specifies a list of Active Directory (AD) users and groups who are entitled to access the desktop pool. The attribute is specified in the form of a Windows Security Identifier (SID) reference. A member value of `<SID=S–1–2–3–4>` represents an AD user or group with the SID value `S–1–2–3–4`. In LDIF format, the left angle (<) character is reserved, so you must place two colons (::) after the attribute name and specify the SID value in base 64 format (for example, `PFNJRD1TLTEtMi0zLTQ+IA==`). Because this attribute is multivalued, you can use it on multiple lines to represent each entry in a list of SIDs. |

In a similar manner to desktop pools, you can create customized LDIF files for other objects that are defined in the LDAP repository, for example:

- Global configuration settings.

- Configuration settings for a specific View Connection Server instance or security server.

- Configuration settings for a specific user.

# Import LDAP Configuration Data

You can import configuration data from an LDIF file into a standard or replica View Connection Server by running the `vdmimport` command.

The `vdmimport` command is available in View 4.5 and later releases. In releases prior to View 4.5, you must use the `LDIFDE` command. See "Import LDAP Configuration Data Using the LDIFDE Command" on page 64.

By default, the `vdmimport` command-line utility is installed in the `C:\Program Files\VMware\VMware View\Server\tools\bin` directory.

The `vdmimport` command does not update, create, or delete any LDAP records that are not defined in the LDIF file. This allows you to customize an LDIF file so that only selected records are affected when you import the file.

You must be logged into a View Connection Server instance as a user in the Administrators role to import configuration data successfully into the View configuration repository.

**To import configuration data into a View Connection Server instance**

1   Select **Start > Command Prompt**.

2   At the command prompt, type the `vdmimport` command and specify an existing LDIF file as an argument to the `–f` option.

    vdmimport –f myexport.LDF

The configuration of your View Connection Server instance is updated with the data from the file, and the number of records that have been successfully updated is displayed. Errors are displayed if some records could not be updated because your role has insufficient privileges.

# Import LDAP Configuration Data Using the LDIFDE Command

You can import configuration data from an LDIF file into a standard or replica View Connection Server by running the Microsoft `LDIFDE` command.

In View 4.5 and later releases, you should use the `vdmimport` command to import configuration data. The `vdmimport` command does not display the large number of error messages that are produced by running the `LDIFDE` command. See "Import LDAP Configuration Data" on page 63.

The `LDIFDE` command does not update, create, or delete any LDAP records that are not defined in the LDIF file. This allows you to customize an LDIF file so that only selected records are affected when you import the file.

For more information about using the LDIFDE command, go to http://support.microsoft.com/kb/237677.

In View 3.1 or earlier, you must be logged into a View Connection Server instance as an administrator, and be a member of the Local Administrators user group.

In View 4.5 and later releases, you must be logged into a View Connection Server instance as a user in the Administrators role to import configuration data successfully into the View configuration repository.

**To import configuration data into a View Connection Server instance using LDIFDE**

1   Select **Start > All Programs > ADAM > ADAM Tools Command Prompt**.

2   At the command prompt, type the `LDIFDE` command and specify an existing LDIF file as an argument to the `-f` option.

    LDIFDE -i -f myexport.LDF -s 127.0.0.1 -z

Enter the other options to the `LDIFDE` command as shown in the example.

The configuration of your View Connection Server instance is updated with the data from the file, and the number of records that have been successfully updated is displayed. A large number of error messages are displayed that indicate whenever an existing entry in the repository has been overwritten. Such errors can be ignored. Errors are also displayed if some records could not be updated because your role has insufficient privileges.

# Integrating with SCOM

<div style="text-align: right; font-size: 3em;">5</div>

You can use Microsoft System Center Operations Manager (SCOM) to monitor the state of View components, including View Connection Server instances and security servers and View services running on these hosts.

This chapter includes the following topics:

## View Management Packs

The View management packs (MPs) require the default System MP that is installed with Microsoft SCOM and the MP for Microsoft Windows Server Base OS System Center Operations Manager 2007.

- `VMware.View.Discovery.mp` contains the agent that discovers instances of View Server installations. See "View Discovery Script" on page 66.

- `VMware.View.Monitoring.mp` contains the views and monitors that you can use with View in the Operations Manager console. See "Views and Monitors" on page 68.

- `VMware.View.Library.mp` contains class and relationship definitions for the managed objects in View. See "Class and Relationship Definitions" on page 71.

- `VMware.View.Image.Library.mp` contains the graphics that represent the classes that are defined in `VMware.View.Library.mp`.

# Name a View Connection Server Group

You can assign a name to a group of one or more View Connection Server hosts and security servers. The Operations Manager console displays this name to help you identify the group within SCOM.

### To name the View Connection Server group

On one of the View Connection Server hosts in the group, use the –C and –c options with the `vdmadmin` command.

`vdmadmin –C –c group_name`

For example, set the name of a View Connection Server group to VCSG01.

`vdmadmin –C –c VCSG01`

# Import the View MPs

When you install the View Connection Server software, the View MPs are also loaded onto the View Connection Server instance or security server. You can copy the View MPs to a SCOM server and use the Import Management Packs wizard from the Operations Manager console to import them.

NOTE   There is a known problem with McAfee VirusScan Enterprise 8.0i blocking the operation of Visual Basic scripts that are used by SCOM. For more information and details of the available patch, go to http://support.microsoft.com/kb/890736/en-us.

### To import the View MPs on a SCOM server

1   Copy the View MPs from their installation directory (usually `C:\Program Files\VMware\VMware View\Server\extras\ManagementPacks`) on a View Connection Server instance or security server to the SCOM server.

2   In the Operations Manager console, right click **Administration\Management Packs**, and select **Import Management Packs**.

3   Use the Import Management Packs wizard to import the View MPs.

4   (Optional) Run the discovery script manually for the systems on which the View software is installed.

More more information, see "Run the Discovery Script" on page 67.

# View Discovery Script

The `VMware.View.Discovery` MP contains a script that finds View installed systems. The discovery script probes the registries of Windows servers for entries that indicate the version of the View software, the type of server, and the name and ID of the View Connection Server group. If the script detects that a View server is installed on a computer, it creates instances of the object classes that are defined in the `VMware.View.Library` MP and establishes the relationships between these managed objects.

The script can only discover a Windows server if you have used the Operations Manager console to enable the proxy agent for the server. See "Enable a Proxy Agent on a Server" on page 70.

The discovery script is scheduled to run once every hour. You can also perform the discovery manually from the Operations Manager console. See "Run the Discovery Script" on page 67.

You can verify the objects that the discovery agent has created for a server by viewing them in the Operations Manager console. See "Display Discovered Objects" on page 67.

The agent discovers the following managed objects for a Connection Server.

- VMware.View.Cluster

- VMware.View.Cluster.Node.Item

- VMware.View.ConnectionServerRole.Item

- VMware.View.Component.ConnectionServer.Item

- VMware.View.Component.Framework.Item

- VMware.View.Component.Web.Item

- VMware.View.Component.Directory.Item

- VMware.View.Component.SecureGateway.Item

- VMware.View.Component.MessageBus.Item

The agent discovers the following managed objects for a Security Server.

- VMware.View.Cluster

- VMware.View.Cluster.Node.Item

- VMware.View.SecurityServerRole.Item

- VMware.View.Component.SecurityServer.Item

- VMware.View.Component.Framework.Item

- VMware.View.Component.SecureGateway.Item

For a description of View object classes and their relationships, see "Class and Relationship Definitions" on page 71.

# Run the Discovery Script

The discovery script is scheduled to run once every hour. You can run the script manually to discover a system that you have added.

**To run the discovery script**

In the Operations Manager console, go to **Monitoring\Windows Computers**, select a computer system, and click the **VMware View Run Discovery Probe** action.

# Display Discovered Objects

You can display the View objects that the discovery script has created for a server.

**To display the objects that the discovery script has created for a server**

In the Operations Manager console, go to **Monitoring\Discovery Inventory**.

# Display Managed Objects

You can display the View objects that SCOM manages, and the relationships between these objects.

**To display managed objects in View and their relationships**

In the Operations Manager console, go to **Monitoring\VMware View**, and select the required view.

# Views and Monitors

The `VMware.View.Monitoring` MP contains the views and monitors that you can use with View in the Operations Manager console. Table 5-1 lists the views that you can use to examine discovered View objects.

**Table 5-1.** Views Available for View

| View | Description |
| --- | --- |
| Active Alerts | Displays critical View alerts. |
| Node State | Displays the state of all discovered members of all View Connection Server groups. |
| Group State | Displays the state of the discovered View Connection Server groups. |
| Groups | Displays a diagram of all discovered View Connection Server groups, members, roles, and components. You can obtain details about objects and their relationships by clicking the icons and the connectors. |
| Connection Server Role Performance Data | Displays the following data sets:<br>■ All Sessions<br>■ All Sessions High<br>■ SVI Sessions<br>■ SVI Sessions High |
| Secure Gateway Role Performance Data | Displays the following data sets:<br>■ Secure Gateway Sessions<br>■ Secure Gateway Sessions High |

You can find the Active Alerts, Group State, Groups, and Node State views under **Monitoring\VMware View** and the Connection Server Role Performance Data and Secure Gateway Role Performance Data views under **Monitoring\VMware View\Performance** in the Operations Manager console.

The `VMware.View.Monitoring` MP provides the following monitor types:

■ Performance monitor – collects system data and return this data to the SCOM performance database and data warehouse. You can examine the data graphically in the Connection Server Role Performance Data and Secure Gateway Role Performance Data views. See "Performance Data Collection Settings" on page 69 and "Display Performance Data" on page 70.

■ Service component monitors – collect information about the state of the View component services. If a monitored service is not running, SCOM sets its state to error and raises an alert. If a component is in the error state, the affected View Connection Server group and its members also enter the error state. See "Display Information About an Alert" on page 70, "Restart a Service" on page 70, and "Close Alerts" on page 71.

Table 5-2 shows the service component monitors that are provided for a View Connection Server instance.

**Table 5-2.** View Service Component Monitors for a View Connection Server instance

| Monitor | Display Name | Monitored Service |
| --- | --- | --- |
| ConnectionServerServiceCheck | Connection Server Service Health | VMware View Connection Server |
| FrameworkServiceCheck | Base Framework Service Health | VMware View Framework Component |
| MessageBusServiceCheck | Message Bus Service Health | VMware View Message Bus Component |

**Table 5-2.** View Service Component Monitors for a View Connection Server instance (Continued)

| Monitor | Display Name | Monitored Service |
|---|---|---|
| SecureGatewayCheck | Security Gateway Service Health | VMware View Security Gateway Component |
| WebServiceCheck | Web Service Health | VMware View Web Component |
| DirectoryServiceCheck | Directory Service Health | VMwareVDMDS |

Table 5-3 shows the service component monitors that are provided for a security server.

**Table 5-3.** View Service Component Monitors for a Security Server

| Monitor | Display Name | Monitored Service |
|---|---|---|
| SecureGatewayServerServiceCheck | Security Server Service Health | VMware View Security Server |
| FrameworkServiceCheck | Base Framework Service Health | VMware View Framework Component |
| SecureGatewayCheck | Security Gateway Service Health | VMware View Security Gateway Component |

- Domain connectivity monitor – verifies that a Connection Server can bind to all the domains of which it is a member. The monitor queries the status of the Web component on a Connection Server every 3 minutes. If a Connection Server cannot bind to a domain, SCOM sets its state to error and raises an alert. See "Display Information About an Alert" on page 70, "Exclude a Domain from Connectivity Monitoring" on page 70, and "Close Alerts" on page 71.

- Event database connectivity monitor – checks that the event database is configured and that events are writable to the database. The monitor queries the Web component every 3 minutes for this information and raises an alert if the event database is not connected.

- Virtual Center (vCenter) connectivity monitor – checks that a View Connection Server instance can connect to the configured vCenter servers. The monitor queries the Web component every 3 minutes for this information and raises an alert if a vCenter server is not available.

## Performance Data Collection Settings

To save storage space, the `System.Performance.OptimizedDataProvider` module is configured to store performance data in the SCOM database when the value of a sampled item has changed by more than a specified tolerance. For more information, see http://msdn.microsoft.com/en-us/library/ee809318.aspx.

Table 5-4 shows the performance data collection settings for the data sets.

**Table 5-4.** Performance Data Collection Settings

| Data set | Frequency | Tolerance | Maximum Sample Separation |
|---|---|---|---|
| All Sessions | 15 minutes | 5% | 4 |
| All Sessions High | 60 minutes | 5% | 4 |
| Secure Gateway Sessions | 15 minutes | 5% | 4 |
| Secure Gateway Sessions High | 60 minutes | 5% | 4 |
| SVI Sessions | 15 minutes | 5% | 4 |
| SVI Sessions High | 60 minutes | 5% | 4 |

# Enable a Proxy Agent on a Server

You must enable the proxy agent on each View Connection Server host or security server that you want to monitor.

**To enable the proxy agent on a server**

1   In the Operations Manager console, go to **Administration\Agent Managed**, select the server, and click **Properties**.

2   Under the **Security** tab, select the option **Allow this agent to act as a proxy and discover managed objects on other computers**.

3   Click **OK**.

# Display Performance Data

You can display graphical information about the performance of a View Connection Server instance or security server.

**To display performance data for a View Connection Server instance or security server**

In the Operations Manager console, go to **Monitoring\VMware View\Performance**, select the **Connection Server Role Performance Data** or **Secure Gateway Role Performance Data** view, and select the required data sets.

# Display Information About an Alert

You can use the Health Explorer to display information about alerts that the health monitors have raised for a View Connection Server group.

**To display active alerts for View components**

1   In the Operations Manager console, go to **Monitoring\VMware View** and select the **Active Alerts** view.

2   Select an alert (⊗) to display the knowledge article for that alert.

# Restart a Service

The service component monitors alert you if a View component service stops working. You can restart a service from the Health Explorer.

**To restart a service following an alert**

1   In the Operations Manager console, go to **Monitoring\VMware View** and select the **Group State** view or the **Group Node State** view.

2   Right click on a View Connection Server group or member that is in the alert state (⊗) and select **Open > Health Explorer**.

3   In the Health Explorer, select the alert and click **Restart the service** in the knowledge article.

# Exclude a Domain from Connectivity Monitoring

The Domain Connectivity Health monitor checks the connectivity between a View Connection Server host's domain and any trusted domains. To avoid seeing alerts for a domain, you can exclude the domain from being monitored.

**To exclude a domain from being monitored for connectivity**

1   In the Operations Manager console, go to **Monitoring\VMware View**, and select the **Group State** view or the **Group Node State** view.

2   Right click on the connection server, and select **Open > Health Explorer**.

3    Right click on the **Domain Connectivity Health** entry for the Connection Server in the Health Explorer and select **Monitor Properties**.

4    Under the **Overrides** tab, click **Override** and select the option for all objects of the same class.

5    In the Override Properties window, select the **Override** check box for the DomainExcludeList parameter, enter the name of the excluded domain in the **Override Setting** field, and select the **Enforced** check box. If you want to exclude more than one domain, use spaces to separate the domain names.

6    Click **Apply** and **OK**.

## Close Alerts

You can close an alert without taking any action. This method does not prevent the alert from being raised again if the underlying cause persists.

**To close alerts**

1    In the Operations Manager console, go to **Monitoring\VMware View** and select the **Active Alerts** view.

2    Select the alert and click the **Close Alert** action.

## Class and Relationship Definitions

The `VMware.View.Library` MP contains the class and relationship definitions for the View MPs. A class can have properties, such as a name or an ID. The relationships between classes describe their hierarchy. For example, the relationship `contains` exists between `VMware.View.Clusters` and `VMware.View.Cluster`, and between `VMware.View.Cluster` and `VMware.View.Cluster.Node`

Table 5-5 shows the View Connection Server group classes that are defined in the `VMware.View.Library` MP.

**Table 5-5.**  View Library View Connection Server Group Classes

| Class Name | Description |
| --- | --- |
| `VMware.View.Cluster` | Represents a View Connection Server group. This class has the properties `ClusterID` and `DisplayName` (the name of the group). |
| `VMware.View.Clusters` | Represents a singleton class that contains instances of `VMware.View.Cluster`. |

Table 5-6 shows the abstract base classes that are defined in the `VMware.View.Library` MP.

**Table 5-6.**  View Library Base Classes

| Class Name | Description |
| --- | --- |
| `VMware.View.Cluster.Node` | Represents a member of a View Connection Server group. This class has the properties `ClusterID`, `ClusterName`, `ProductVersion`, and `InstallPath`. |
| `VMware.View.Component` | Represents a View component that has been installed on a member of a View Connection Server group. This class has the property `Name`. |
| `VMware.View.Component.ConnectionServer` | Represents the Connection Server component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component`. |
| `VMware.View.Component.Directory` | Represents the Directory component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component`. |

**Table 5-6.** View Library Base Classes (Continued)

| Class Name | Description |
| --- | --- |
| `VMware.View.Component.Framework` | Represents the Framework component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component`. |
| `VMware.View.Component.MessageBus` | Represents the Message Bus component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component`. |
| `VMware.View.Component.SecurityGateway` | Represents the Security Gateway component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component`. |
| `VMware.View.Component.SecurityServer` | Represents the Security Server component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component`. |
| `VMware.View.Component.Web` | Represents the Web component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component`. |
| `VMware.View.ConnectionServerRole` | Represents a member of a View Connection Server group with the Connection Server installed on it. This class inherits its properties from `VMware.View.NodeRole`. |
| `VMware.View.NodeRole` | Represents the role of a member of a View Connection Server group. |
| `VMware.View.SecurityServerRole` | Represents a member of a View Connection Server group with the Security Server installed on it. This class inherits its properties from `VMware.View.NodeRole`. |

The currently supported instances that are derived from these classes must be View version 5.1.$x$ or a later release. Table 5-7 shows the concrete classes that are defined in the `VMware.View.Library` MP.

**NOTE**   These concrete classes are the latest versions and are supported in View 5.1.$x$ and later releases.

**Table 5-7.** View Library Concrete Classes

| Class Name | Description |
| --- | --- |
| `VMware.View.Cluster.Node.Item` | Represents a View Connection Server group member that has version 5.1.$x$ or a later release of View installed. This class inherits its properties from `VMware.View.Cluster.Node`. |
| `VMware.View.Component.ConnectionServer.Item` | Represents version 5.1.$x$ or a later release of the Connection Server component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component.Connection Server`. |
| `VMware.View.Component.Directory.Item` | Represents version 5.1.$x$ or a later release of the Directory component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component.Directory`. |
| `VMware.View.Component.Framework.Item` | Represents version 5.1.$x$ or a later release of the Framework component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component.Framework`. |
| `VMware.View.Component.MessageBus.Item` | Represents version 5.1.$x$ or a later release of the Message Bus component that has been installed on a member of a View Connection Server group. This class inherits its properties from `VMware.View.Component.MessageBus`. |

**Table 5-7.** View Library Concrete Classes (Continued)

| Class Name | Description |
|---|---|
| VMware.View.Component.SecurityGateway.Item | Represents version 5.1.*x* or a later release of the Security Gateway component that has been installed on a member of a View Connection Server group. This class inherits its properties from VMware.View.Component.SecureGateway. |
| VMware.View.Component.SecurityServer.Item | Represents version 5.1.*x* or a later release of the Security Server component that has been installed on a member of a View Connection Server group. This class inherits its properties from VMware.View.Component.SecurityServer. |
| VMware.View.Component.Web.Item | Represents version 5.1.*x* or a later release of the Web component that has been installed on a member of a View Connection Server group. This class inherits its properties from VMware.View.Component.Web. |
| VMware.View.ConnectionServerRole.Item | Represents a member of a View Connection Server group with version 5.1.*x* or a later release of the Connection Server installed on it. This class inherits its properties from VMware.View.NodeRole. |
| VMware.View.SecurityServerRole.Item | Represents a member of a View Connection Server group with version 5.1.*x* or a later release of the Security Server installed on it. This class inherits its properties from VMware.View.NodeRole. |

The VMware.View.Library MP also contains friendly name strings for classes and properties. The SCOM console displays friendly names in preference to class and property names.

# Examining PCoIP Session Statistics

<span style="font-size:large;">**6**</span>

You can use Windows Management Instrumentation (WMI) to examine performance statistics for a PCoIP session by using any of the supported programming interfaces, including C#, C++, PowerShell, VBScript, VB .NET, and Windows Management Instrumentation Command-line (WMIC). You can also use the Microsoft WMI Code Creator tool to generate VBScript, C#, and VB .NET code that accesses the PCoIP performance counters. For more information about WMI, WMIC, and the WMI Code Creator tool, go to http://technet.microsoft.com/en-us/library/bb742610.aspx and http://www.microsoft.com/downloads/en/details.aspx?familyid=2cc30a64-ea15-4661-8da4-55bbc145c30e&displaylang=en.

This chapter includes the following topics:

## Usage Notes for Recorded Statistics

The WMI namespace for the PCoIP session statistics is `root\CIMV2`.

Administrator privileges are required to access the performance counters remotely.

The names of the statistics are suffixed with `(Server)` or `(Client)`, according to whether the statistic is recorded on the PCoIP server or on the PCoIP client.

You can use Windows Performance Monitor (PerfMon) with the counters to calculate averages over a specified sampling period.

All statistics are reset to 0 when a PCoIP session is closed. If the WMI property `SessionDurationSeconds` is a non-zero value and stays constant, the PCoIP server was forcefully ended or crashed. If the `SessionDurationSeconds` property changes from a non-zero value to 0, the PCoIP session is closed.

To avoid a division-by-zero error, verify that the denominator in the expressions for calculating bandwidth or packet-loss percentage does not evaluate to zero.

USB statistics are recorded for zero clients, but not for thin clients or software clients.

# General Session Statistics

The WMI class name for PCoIP general session statistics is
`Win32_PerfRawData_TeradiciPerf_PCoIPSessionGeneralStatistics`.

Table 6-1 shows the general statistics for a PCoIP session.

**Table 6-1.** General Statistics

| WMI Property Name | Description |
| --- | --- |
| BytesReceived | Total number of bytes of PCoIP data that have been received since the PCoIP session started. |
| BytesSent | Total number of bytes of PCoIP data that have been transmitted since the PCoIP session started. |
| PacketsReceived | Total number of packets that have been received successfully since the PCoIP session started. Not all packets are the same size. |
| PacketsSent | Total number of packets that have been transmitted since the PCoIP session started. Not all packets are the same size. |
| RXPacketsLost | Total number of received packets that have been lost since the PCoIP session started. |
| SessionDurationSeconds | Total number of seconds that the PCoIP Session has been open. |
| TXPacketsLost | Total number of transmitted packets that have been lost since the PCoIP session started. |

To calculate the bandwidth in kilobits per second for received PCoIP data over the time interval from time t1 to time t2:

```
(BytesReceived[t2]–BytesReceived[t1]) * 8 / (1024 * (t2–t1))
```

To calculate the bandwidth in kilobits per second for transmitted PCoIP data over the time interval from time t1 to time t2:

```
(BytesSent[t2]–BytesSent[t1]) * 8 / (1024 * (t2–t1))
```

To calculate the percentage of received packets that are lost:

```
100 / (1 + ((PacketsReceived[t2]–PacketsReceived[t1])/(RXPacketsLost[t2]–RXPacketsLost[t1])))
```

To calculate the percentage of transmitted packets that are lost:

```
100 * (TXPacketsLost[t2]–TXPacketsLost[t1]) / (PacketsSent[t2]–PacketsSent[t1])
```

# Audio Statistics

The WMI class name for PCoIP audio statistics is
`Win32_PerfRawData_TeradiciPerf_PCoIPSessionAudioStatistics`.

The audio statistics do not include audio data that is carried within USB data.

Table 6-2 shows the audio statistics for a PCoIP session.

**Table 6-2.** Audio Statistics

| WMI Property Name | Description |
| --- | --- |
| AudioBytesReceived | Total number of bytes of audio data that have been received since the PCoIP session started. |
| AudioBytesSent | Total number of bytes of audio data that have been sent since the PCoIP session started. |
| AudioRXBWkbitPersec | Bandwidth for ingoing audio packets averaged over the sampling period, in seconds. |

**Table 6-2.** Audio Statistics (Continued)

| WMI Property Name | Description |
|---|---|
| AudioTXBWkbitPersec | Bandwidth for outgoing audio packets averaged over the sampling period, in seconds. |
| AudioTXBWLimitkbitPersec | Transmission bandwidth limit in kilobits per second for outgoing audio packets. The limit is defined by a GPO setting. |

To calculate the bandwidth in kilobits per second for received audio data over the time interval from time t1 to time t2:

    (AudioBytesReceived[t2]−AudioBytesReceived[t1]) * 8 / (1024 * (t2−t1))

To calculate the bandwidth in kilobits per second for transmitted audio data over the time interval from time t1 to time t2:

    (AudioBytesSent[t2]−AudioBytesSent[t1]) * 8 / (1024 * (t2−t1))

Do not use AudioTXBWkbitPersec for these calculations.

## Imaging Statistics

The WMI class name for PCoIP imaging statistics is Win32_PerfRawData_TeradiciPerf_PCoIPSessionImagingStatistics.

Table 6-3 shows the imaging statistics for a PCoIP session.

**Table 6-3.** Imaging Statistics

| WMI Property Name | Description |
|---|---|
| ImagingBytesReceived | Total number of bytes of imaging data that have been received since the PCoIP session started. |
| ImagingBytesSent | Total number of bytes of imaging data that have been transmitted since the PCoIP session started. |
| ImagingDecoderCapabilitykbitPersec | Estimated processing capability of the imaging decoder in kilobits per second. This statistic is updated once per second. |
| ImagingEncodedFramesPersec | Number of imaging frames that were encoded over a one-second sampling period. |
| ImagingActiveMinimumQuality | Lowest encoded quality value on a scale from 0 to 100. This statistic is updated once per second. This counter does not correspond to the GPO setting for minimum quality. |
| ImagingRXBWkbitPersec | Bandwidth for incoming imaging packets averaged over the sampling period, in seconds. |
| ImagingTXBWkbitPersec | Bandwidth for outgoing imaging packets averaged over the sampling period, in seconds. |

To calculate the bandwidth in kilobits per second for received imaging data over the time interval from time t1 to time t2:

    (ImagingBytesReceived[t2]−ImagingBytesReceived[t1]) * 8 / (1024 * (t2−t1))

Do not use ImagingRXBWkbitPersec for the preceding calculation.

To calculate the bandwidth in kilobits per second for transmitted imaging data over the time interval from time t1 to time t2:

    (ImagingBytesSent[t2]−ImagingBytesSent[t1]) * 8 / (1024 * (t2−t1))

Do not use ImagingTXBWkbitPersec for the preceding calculation.

# Network Statistics

The WMI class name for PCoIP network statistics is
`Win32_PerfRawData_TeradiciPerf_PCoIPSessionNetworkStatistics`.

Table 6-4 shows the network statistics for a PCoIP session.

**Table 6-4.** Network Statistics

| WMI Property Name | Description |
| --- | --- |
| RoundTripLatencyms | Round trip latency in milliseconds between the PCoIP server and the PCoIP client. |
| RXBWkbitPersec | Overall bandwidth for incoming PCoIP packets averaged over the sampling period, in seconds. |
| RXBWPeakkbitPersec | Peak bandwidth in kilobits per second for incoming PCoIP packets over a one-second sampling period. |
| RXPacketLossPercent | Percentage of received packets lost during a sampling period. |
| TXBWkbitPersec | Overall bandwidth for outgoing PCoIP packets averaged over the sampling period, in seconds. |
| TXBWActiveLimitkbitPersec | Estimated available network bandwidth in kilobits per second. This statistic is updated once per second. |
| TXBWLimitkbitPersec | Transmission bandwidth limit in kilobits per second for outgoing packets. The limit is the minimum of the following values:<br>■ GPO bandwidth limit for the PCoIP client<br>■ GPO bandwidth limit for the PCoIP server<br>■ Bandwidth limit for the local network connection<br>■ Negotiated bandwidth limit for the Zero Client firmware based on encryption limits |
| TXPacketLossPercent | Percentage of transmitted packets lost during a sampling period. |

To calculate the bandwidth in kilobits per second for received data over the time interval from time t1 to time t2:

```
(BytesReceived[t2]–BytesReceived[t1]) * 8 / (1024 * (t2–t1))
```

Do not use `RXBWkbitPersec` for the preceding calculation.

To calculate the bandwidth in kilobits per second for transmitted data over the time interval from time t1 to time t2:

```
(BytesSent[t2]–BytesSent[t1]) * 8 / (1024 * (t2–t1))
```

Do not use `TXBWkbitPersec` for the preceding calculation.

To calculate the packet loss in percentage for received data over the time interval from time t1 to time t2:

```
PacketsReceived during interval = (PacketsReceived[t2]–PacketsReceived[t1])

RXPacketsLost during interval = (RXPacketsLost[t2]–RXPacketsLost[t1])

RXPacketsLost % = RXPacketsLost during interval /
(RXPacketsLost during interval + PacketsReceived during interval) * 100
```

Do not use `RXPacketLostPercent` or `RXPacketLostPercent_Base` for the preceding calculation.

To calculate the packet loss in percentage for transmitted data over the time interval from time t1 to time t2:

```
PacketsSent during interval = (PacketsSent[t2]–PacketsSent[t1])

TXPacketsLost during interval = (TXPacketsLost[t2]–TXPacketsLost[t1])

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

Use this formula to prevent the packet loss percent from becoming greater than 100%. This calculation is required because `PacketsLost` and `PacketsSent are asynchronous`..

Do not use `TXPacketLostPercent` or `TXPacketLostPercent_Base` for the preceding calculation.

## USB Statistics

The WMI class name for PCoIP USB statistics is
`Win32_PerfRawData_TeradiciPerf_PCoIPSessionUSBStatistics`.

Table 6-5 shows the USB statistics for a PCoIP session.

**Table 6-5.** USB Statistics

| WMI Property Name | Description |
| --- | --- |
| USBBytesReceived | Total number of bytes of USB data that have been received since the PCoIP session started. |
| USBBytesSent | Total number of bytes of USB data that have been transmitted since the PCoIP session started. |
| USBRXBWkbitPersec | Bandwidth for incoming USB packets averaged over the sampling period, in seconds |
| USBTXBWkbitPersec | Bandwidth for outgoing USB packets averaged over the sampling period, in seconds. |

To calculate the bandwidth in kilobits per second for received USB data over the time interval from time t1 to time t2:

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

Do not use `USBRXBWkbitPersec` for the preceding calculation.

To calculate the bandwidth in kilobits per second for transmitted USB data over the time interval from time t1 to time t2:

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

Do not use `USBTXBWkbitPersec` for the preceding calculation.

## Examples of Using PowerShell to Examine PCoIP Statistics

Retrieve the PCoIP network statistics for the client cm-02.

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

Retrieve the PCoIP general session statistics for desktop dt-03 if any transmitted packets have been lost.

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

# Dynamically Setting Desktop Policies with Start Session Scripts

# 7

When a user connects or reconnects to a View desktop, you might want to dynamically configure particular View desktop settings before the desktop session starts, according to information recieved from Horizon Client and View Connection Server. For example, you might want to set different policies for clipboard redirect, mapped drives, or other desktop features, depending on the location of the user's client system.

Using information from Horizon Client, such as the client system's IP address, and information from View Connection Server, such as the DNS name of the View Connection Server instance, you can set up scripts that determine the View desktop experience when a user requests a desktop connection. For example, a script might allow mapped drives, clipboard redirection, and so on, for a user with an IP address in your organization's internal domain , but disallow these features for an IP address that comes from an external domain.

Using start session scripts lets you dynamically configure desktop policies based on the client device and the user's location without having to set up multiple desktop pools. You can enforce your organization's security policies without the storage and licensing costs of adding pools to your deployment.

**NOTE**   Use start session scripts only if you have a particular need to configure desktop policies before a desktop session begins. As a best practice, use the View Agent `CommandsToRunOnConnect` and `CommandsToRunOnReconnect` group policy settings to run command scripts after a desktop session is connected or reconnected. Running scripts within a desktop sesion will satisfy most use cases. For details, see "Running Commands on View Desktops" in the *View Administration* document.

This chapter discusses the following topics:

- "Overview of Configuring Start Session Scripts" on page 81

- "Example Start Session Scripts" on page 82

- "Enable the VMware View Script Host Service" on page 83

- "Add Windows Registry Entries for a Start Session Script" on page 83

## Overview of Configuring Start Session Scripts

To enable a start session script, you perform the following tasks on the View desktop:

- Enable the VMware View Script Host service

- (Optional) Configure the account of the user who has the appropriate privileges to run the script

- Add entries in the Windows registry to enable the script

You must configure all View desktops that need to run start session scripts. There is no mechanism to propagate registry changes, VMware View Script Host service configuration changes, and start session scripts to multiple View desktop virtual machines.

# Example Start Session Scripts

Start session scripts cannot run interactively. A start session script runs in an environment created by View, and the script must obtain its input data from this environment. The input data is gathered from environment variables on the client computer.

For a list of similar variables, see "Client System Information Sent to View Desktops" in the *Setting Up Desktop and Application Pools in View* document. Note that start session variables have the prefix `VDM_StartSession_` instead of `ViewClient_`. For example, the start session variable containing the client system's IP address is `VDM_StartSession_IP_Address`.

You should ensure that a start session script validates the existence of any environment variable that it uses.

The following sample Visual Basic script writes all the environment variables being provided to the script into a file, so that you can see example data in your own environment:

```
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
```

You might save the preceding sample script as `c:\sample.vbs`.

The following sample script tests the timeout functionality:

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

The following sample script tests a non-zero exit code:

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

**NOTE**   Make sure your start session scripts run quickly.

If you set the `WaitScriptsOnStartSession` value in the Windows registry, your start session script must finish running before View Agent can respond to the StartSession message sent from View Connection Server. A long-running script is likely to cause the StartSession request from View Connection Server to time out. If a timeout occurs, View Connection Server tries to connect the user to another virtual machine (if the pool uses floating assignments) or, if no virtual machine is available, to reject the user's connection request.

In addition, you should set a hard timeout for the script host operation so that, if a script does run too long, a specific error can be returned.

# Enable the VMware View Script Host Service

You must configure the VMware View Script Host service to be enabled on each View desktop virtual machine where you want View to run the start session script. By default, the VMware View Script Host service is disabled.

**To configure the VMware View Script Host service**

1   Start the Windows Services tool by entering `services.msc` at the command prompt.

2   In the details pane, right-click on the VMware View Script Host service entry and select **Properties**.

3   On the **General** tab, in **Startup type**, select **Automatic**.

4   If you do not want the local system account to run the start session script, select **This account**, and enter the details of the account to run the start session script.

5   Click **OK** and exit the Services tool.

## Configure a User Account to Run the Start Session Script

Optionally, you can change the user account under which the start session scripts run. Start session scripts run in the context of the VMware View Script Host service. By default, this service is configured to run as the SYSTEM user.

It is important to remember that start session scripts are run outside a desktop user session and not by the desktop user account. Information is sent directly from the client computer within a script running as the SYSTEM user.

Make sure that the path where you configure your start session scripts is accessible only to the SYSTEM account and to local administrators. (Set the ACL for the base key to be accessible to these accounts only.)

As a best practice, place the start session scripts in the following location:

`View_Agent_install_path\scripts`

For example: `%ProgramFiles%\VMware\VMware View\Agent\scripts\sample.vbs`

By default, this folder is accessible only by the SYSTEM and administrator accounts.

# Add Windows Registry Entries for a Start Session Script

To make a start session script available for execution, you must add the path to the script in the Windows registry. You also must add and enable a start session value, `RunScriptsOnStartSession`, in the registry. To prevent scripts from timing out, you should set a registry value to specify timeout values in seconds rather than minutes. Finally, you can set additional, optional entries to govern timeout values.

You must create these registry entries on each View desktop virtual machine where you want to run a start session script.

**To add Windows registry entries to enable a start session script on a View desktop virtual machine**

1   Start the Windows Registry Editor by entering `regedit` at the command prompt.

2   In the registry, navigate to `HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents`.

3   In the navigation area, right-click **ScriptEvents**, select **New > Key**, and create a key named StartSession.

4   In the navigation area, right-click **StartSession**, select **New > String Value**, and create a string value that identifies the start session script to be run. For example: `SampleScript`

    To run more than one start session script, create a string value entry for each script under the **StartSession** key. You cannot specify the order in which these scripts run. If the scripts must run in a specified order, invoke them from a single control script.

5   In the topic area, right-click on the entry for the new string value and select **Modify**.

6   In **Value data**, enter the command line that invokes the start session script and click **OK**.

    Specify the full path of the start session script and any files that it requires.

7   Navigate to `HKLM\SOFTWARE\VMware, Inc.\VMware VDM\Agent\Configuration`.

    If a **Configuration** key does not exist, right-click **Agent**, select **New > Key**, and create the key.

8   In the navigation area, right-click **Configuration**, select **New > DWord (32 bit) Value**, and type `RunScriptsOnStartSession`.

9   In the topic area, right-click on the entry for the new DWord value and select **Modify**.

10  In **Value data**, type 1 to enable start session scripting. Type 0 to disable this feature. 0 is the default value.

11  (Optional) To delay the StartSession response by View Agent, add a second DWord value to the **Configuration** key called `WaitScriptsOnStartSession`.

    A `WaitScriptsOnStartSession` data value of 1 causes View Agent to delay sending a StartSession response and fail if the scripts do not complete. A value of 0 means that View Agent does not wait for the scripts to complete or check script exit codes before sending the StartSession response. The default value is 0.

12  Set the timeout values in seconds rather than minutes.

    a   Navigate to `HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents`.

    b   Add a DWord value called `TimeoutsInMinutes`.

    c   Set a data value of 0.

    Setting this timeout value in seconds allows you to configure the VMware View Script Host service timeout value in seconds. For example, if you set the Script Host service timeout to 30 seconds, you can ensure that a start session script will either finish running or time out before a View Connection Server timeout occurs.

13  (Optional) To enable the VMware View Script Host service to time out the start session script, navigate to `HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents\StartSession`. Right click the **Default** (@) key, select **Modify**, enter the timeout value in milliseconds, and click **OK**. A value of 0 means that no timeout is set.

14  Exit the Registry Editor and restart the system.

# Index