

View Integration

VMware Horizon 6

Version 6.2

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see <http://www.vmware.com/support/pubs>.

EN-001913-00

vmware[®]

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

Copyright © 2015 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

View Integration	5
1 Introduction to View Integration	7
View Components	7
Integration Interfaces to View	8
2 Integrating View with the Event Database	9
Event Database Tables and Schemas	9
Connection Broker Events	11
View Agent Events	16
View Administrator Events	17
Event Message Attributes	24
Sample Database Queries and Views	25
3 Using View PowerCLI	27
Getting Started with View PowerCLI	27
View Administrator, View PowerCLI, and vdmadmin Compared	30
View PowerCLI cmdlet Reference	32
View PowerCLI cmdlet Parameters	34
Examples of Using View PowerCLI cmdlets	38
Examples of Using View PowerCLI to Perform Advanced Tasks	45
Assigning Multiple Network Labels to a Desktop Pool	50
4 Customizing LDAP Data	59
Introduction to LDAP Configuration Data	59
Modifying LDAP Configuration Data	60
5 Integrating View with Microsoft SCOM	65
Setting Up a SCOM Integration	65
Monitoring View in the Operations Manager Console	70
6 Examining PCoIP Session Statistics with WMI	75
Using PCoIP Session Statistics	75
General PCoIP Session Statistics	76
PCoIP Audio Statistics	76
PCoIP Imaging Statistics	77
PCoIP Network Statistics	78
PCoIP USB Statistics	79
Examples of Using PowerShell cmdlets to Examine PCoIP Statistics	80

7	Setting Desktop Policies with Start Session Scripts	81
	Obtaining Input Data for a Start Session Script	81
	Best Practices for Using Start Session Scripts	81
	Preparing a View Desktop to Use a Start Session Script	82
	Sample Start Session Scripts	84
	 Index	 87

View Integration

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

Introduction to View Integration

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 7
- [“Integration Interfaces to View,”](#) on page 8

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 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 RDS host.
View Agent	A software service that is installed on all guest virtual machines, physical systems, or RDS hosts to allow them to be managed by View. View 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 enable users to connect to their desktops.
View Administrator	A Web application that enables View administrators to configure View Connection Server, deploy desktop and application pools, manage machines, control user authentication, initiate and examine system events, and perform 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 instance 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 instance to enable 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 View 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 end-user actions, administrator actions, alerts that report system failures and errors, and statistical sampling.

End-user actions include logging and starting desktop and application sessions. Administrator actions include adding entitlements and creating desktop and application pools. An example of statistical sampling is 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 9
- [“Connection Broker Events,”](#) on page 11
- [“View Agent Events,”](#) on page 16
- [“View Administrator Events,”](#) on page 17
- [“Event Message Attributes,”](#) on page 24
- [“Sample Database Queries and Views,”](#) on page 25

Event Database Tables and Schemas

View uses database tables to implement the event database. The event database prepends the names of these tables with a prefix that you define when you set up the database.

Event Database Tables

The following table shows the database 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.

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.

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.

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.

For information about setting up the event database, see the *View Installation* document.

NOTE Events might 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>.

Event Database Schemas

The following table shows the schema for the event and event_historical database tables.

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

Column Name	Oracle Data Type	SQL Server 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.

Table 2-2. Schema for the event and event_historical Tables (Continued)

Column Name	Oracle Data Type	SQL Server Data Type	Description
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.

The following table shows the schema for the event_data and event_data_historical database tables.

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

Column Name	Oracle Data Type	SQL Server Data Type	Description
BooleanValue	SMALLINT	tinyint	Value of a Boolean attribute. <ul style="list-style-type: none"> ■ 0 = false ■ 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. <ul style="list-style-type: none"> ■ 0 = StrValue ■ 1 = IntValue ■ 2 = TimeValue ■ 3 = BooleanValue

Connection Broker Events

Connection broker events report View Connection Server-related information, such as desktop and application sessions, user authentication failures, and provisioning errors.

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 the vCenter driver that View uses to track a vCenter Server instance.

The `BROKER_VC_STATUS_*` events report the state of a vCenter Server instance.

The following table lists all the event types for View Connection Server.

Table 2-4. Connection Broker Events

Event Type	Severity	ModuleAndEventText
BROKER_AGENT_OFFLINE	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}
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)

Event Type	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}
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_LCReservation_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_LCReservation_SET	ERROR	Provisioning error occurred on Pool \${DesktopId} because available free disk space is reserved for linked clones

Table 2-4. Connection Broker Events (Continued)

Event Type	Severity	ModuleAndEventText
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_COMPOSE_AGENT_INIT_FAILED	ERROR	Provisioning error occurred for Machine \${MachineName}: View Composer agent initialization failed
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

Table 2-4. Connection Broker Events (Continued)

Event Type	Severity	ModuleAndEventText
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 \${SVIPPath}
BROKER_SVI_ARCHIVE_UDD_SUCCEEDED	AUDIT_SUCCESS	Archived user data disk \${UserDiskName} to location \${SVIPPath}
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
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

Table 2-4. Connection Broker Events (Continued)

Event Type	Severity	ModuleAndEventText
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}
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

View Agent events report View Agent-related information, such as the users who have logged in to or disconnected from a specific machine, whether View Agent has shut down on a specific machine, and whether View Agent has sent a start up message from a specific machine to View Connection Server.

Table 2-5. View Agent Events

Event Type	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

View Administrator events report information about actions that users initiate in 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}
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_ADMINISTRATOR_REMOVE_FAILED	AUDIT_FAIL	#{UserDisplayName} failed to remove all permissions for Administrator #{AdminPermissionEntity}
ADMIN_ADMINISTRATOR_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

Table 2-6. View Administrator Events (Continued)

EventType	Severity	ModuleAndEventText
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}
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

Table 2-6. View Administrator Events (Continued)

EventType	Severity	ModuleAndEventText
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
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})

Table 2-6. View Administrator Events (Continued)

EventType	Severity	ModuleAndEventText
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}
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})

Table 2-6. View Administrator Events (Continued)

EventType	Severity	ModuleAndEventText
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_SENDSMSG	AUDIT_SUCCESS	\${UserDisplayName} sent message (\${SessionMessage}) to session (User \${UserName}, Desktop \${MachineName})
ADMIN_SESSION_SENDSMSG_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})
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}

Table 2-6. View Administrator Events (Continued)

EventType	Severity	ModuleAndEventText
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}
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}

Table 2-6. View Administrator Events (Continued)

EventType	Severity	ModuleAndEventText
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}
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

ModuleAndEventText messages use certain attributes. To determine the data type for an attribute, you can examine its value in the type column in the event_data or event_data_historical table.

Table 2-7. Attributes that ModuleAndEventText Messages Use

Attribute Name	Description
AdminFolderName	Name of a folder that requires privileged access.
AdminPermissionEntity	Name of an object that requires 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.
EntitlementDisplay	Display name of a desktop entitlement.
MachineId	Name 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.

Table 2-7. Attributes that ModuleAndEventText Messages Use (Continued)

Attribute Name	Description
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

You can query the event_historical database to display error events, warning events, and specific recent events.

NOTE Replace the dbo.VE_ prefix in the following examples with the appropriate prefix for your event database.

List Error Events

The following query displays 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'
);
```

List Warning Events

The following query displays 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 Recent Events

The following query lists 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'
);
```

The following query lists 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'
);
```

The following query lists 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'
);
```

The following query lists 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'
);
```

The following query lists 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

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

This chapter includes the following topics:

- [“Getting Started with View PowerCLI,”](#) on page 27
- [“View Administrator, View PowerCLI, and vdmadmin Compared,”](#) on page 30
- [“View PowerCLI cmdlet Reference,”](#) on page 32
- [“View PowerCLI cmdlet Parameters,”](#) on page 34
- [“Examples of Using View PowerCLI cmdlets,”](#) on page 38
- [“Examples of Using View PowerCLI to Perform Advanced Tasks,”](#) on page 45
- [“Assigning Multiple Network Labels to a Desktop Pool,”](#) on page 50

Getting Started with View PowerCLI

PowerShell is a command line and scripting environment designed for Microsoft Windows. PowerShell uses the .NET object model and provides administrators with management and automation capabilities. You work with PowerShell by running commands, which are called cmdlets in PowerShell. The command line syntax for View PowerCLI cmdlets is the same as generic PowerShell syntax.

The View PowerCLI cmdlets are defined in the `PowershellServiceCmdlets.dll` file, which is installed in the `C:\Program Files\VMware\VMware View\Server\bin` directory. The `PowershellServiceCmdlets.dll` file constitutes the `VMware.View.Broker` snapin.

You can edit and extend the View PowerCLI cmdlets script configuration file, `InitViewCmdlets.ps1`, to define cmdlet aliases, configure the environment, and set startup actions. `InitViewCmdlets.ps1` is in the `Extras` folder in the View installation directory.

You can use View PowerCLI cmdlets in conjunction with vSphere PowerCLI cmdlets. vSphere PowerCLI cmdlets provide an administrative interface to VMware vSphere. If vSphere PowerCLI is installed on a View Connection Server instance, the vSphere PowerCLI cmdlets load when you launch View PowerCLI.

You can refer to virtual machines and vCenter Server instances by ID in View PowerCLI, but you cannot pass these entities as objects. For other vSphere objects, such as resource pools and folders, you must provide a full path. You can use View PowerCLI cmdlets to examine the configuration of vCenter Server instances within View.

For general information about using PowerShell, see the Microsoft documentation.

Start the PowerShell Console with View PowerCLI Loaded

You can run the View PowerCLI cmdlets directly on a View Connection Server host.

Prerequisites

Verify that the View Connection Server host has the following software installed.

- View 4.5 or later
- Microsoft .NET framework
- Windows PowerShell 1.0

Procedure

- 1 Log in to the View Connection Server instance as a user in a role that has sufficient privileges to modify configuration data.

For example, the Administrators role can modify configuration data. A read-only role cannot update configuration data.

- 2 Select **Start > All Programs > VMware > View PowerCLI**.

What to do next

If you receive an error message that states the script configuration file cannot be loaded because the execution of scripts is disabled, type the PowerShell `Set-ExecutionPolicy Unrestricted` command and restart the PowerShell console.

Use View PowerCLI cmdlets from a Remote System

You can use the PowerShell remoting feature to access View PowerCLI cmdlets from a remote system.

Procedure

- 1 On the remote system, open the `C:\Windows\System32\WindowsPowerShells\v1.0\Profile.ps1` file in a text editor.
- 2 Add the line `add-pssnapin vm*` to the `Profile.ps1` file.
- 3 Save your changes.

The View PowerCLI snapin to the PowerShell profile is added on the remote system.

What to do next

Take the same precautions for protecting the View PowerCLI operations as you would for other remoting PowerShell operations.

Display Help for View PowerCLI

You can display help for View PowerCLI by typing commands in the PowerShell console.

Procedure

- 1 On a View Connection Server instance, select **Start > All Programs > VMware > View PowerCLI**.

- 2 Display the View PowerCLI help.

Option	Action
List all View PowerCLI cmdlets	Run the <code>Get-Command</code> cmdlet. For example: <code>Get-Command -PSSnapin VMware.View.Broker more</code>
Display help for a specific cmdlet	Type <code>Get-Help</code> followed by the name of the cmdlet. For example: <code>Get-Help Add-ViewVC more</code>
Display detailed help for a specific cmdlet	Type <code>Get-Help</code> followed by the name of the cmdlet and the <code>-full</code> parameter. For example: <code>Get-Help Add-ViewVC -full more</code> Alternatively, use the <code>help</code> alias for <code>Get-Help</code> . For example: <code>Add-ViewVC -full more</code>

Examining View PowerCLI cmdlet Errors

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 cause of an error.

To control how PowerShell handles non-terminating errors and how it displays errors in the shell, set the standard PowerShell `$ErrorActionPreference` and `$ErrorView` automatic variables.

Piping and Specifying Objects of the Same Type

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 produces this error.

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

Escaping Characters in vCenter Server Path Names

If you specify a path to a vCenter Server folder that includes certain special characters in the name of an entity, you must escape the 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, represent the path to the folder `/datacenter_01/vm/img%-12` as `/datacenter_01/vm/img%25-12`.

Certain cmdlets and parameters require escape sequences in entity names.

Table 3-2. cmdlet Parameters that Require Escape Sequences

cmdlet	Parameters that Require Escape Sequences
Add-AutomaticLinkedClonePool	-datastoreSpecs
Update-AutomaticLinkedClonePool	-parentVMPath -resourcePoolPath -vmfolderPath
Add-AutomaticPool	-datastorePaths
Update-AutomaticPool	-resourcePoolPath -templatePath -vmfolderPath
Send-LinkedCloneRecompose	-parentVMPath

View Administrator, View PowerCLI, and vdmadmin Compared

You can use View Administrator, View PowerCLI cmdlets, and vdmadmin commands to perform administrative operations on View objects. Not all administrative operations are available in all utilities.

Table 3-3. View Administrator, View PowerCLI, and vdmadmin Operations

Object	Operation	View Administrator	View PowerCLI	vdmadmin
Desktop pool	Add	X	X	
	Assign dedicated			X
	Assign ThinApp	X		
	Disable	X	X	
	Enable	X	X	
	Entitle user	X	X	
	Get information	X	X	
	Get unentitled policies	X		X
	Get unentitled users	X		X
	Remove	X	X	
	Remove assignment			X
	Remove entitlement	X	X	
	Restrict entitlement	X		
	Set policy	X	X	
	Update	X	X	
	Assign network label		X	
	Get network label configuration		X	
Domain filter	Get information			X
	Remove filter			X
	Set filter			X
Events	Get list	X	X	X
	Get report		X	X
Folder	Add	X		
	Get information	X		

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

Object	Operation	View Administrator	View PowerCLI	vdmadmin
	Move	X		
	Remove	X		
Kiosk mode	Add client account			X
	Disable authentication			X
	Enable authentication			X
	Get information			X
	Get defaults			X
	Remove client account			X
	Set defaults			X
Linked-clone desktop	Rebalance	X	X	
	Recompose	X	X	
	Recreate	X		
	Refresh	X	X	
	Restore	X		
	Set storage overcommit	X	X	
	Get network label configuration		X	
Permission	Add	X		
	Get information	X		
	Remove	X		
Persistent user data disk	Attach	X		
	Delete	X		
	Detach	X		
	Get information	X	X	
	Replace	X		
Physical computer (View Agent installed)	Get information	X	X	X
Remote session	Disconnect	X	X	
	Get information	X	X	
	Log out	X	X	
Role	Add	X		
	Modify	X		
	Remove	X		
RDS host	Get information	X	X	
User	Configure policy	X		
	Create administrator	X		
	Get information	X	X	X
	Remove administrator	X		
	Update FSP			X

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

Object	Operation	View Administrator	View PowerCLI	vdmadmin
vCenter Server instance	Add	X	X	
	Get information	X	X	
	Remove	X	X	
	Update	X	X	
View Agent	Create DCT bundle			X
	Get copy of log file			X
	Get list of log files			X
	Get logging level			X
	Get status			X
	Get version			X
	Override IP address			X
	Set logging level			X
View Composer domain	Get information		X	
View Connection Server instance	Back up configuration	X		
	Get information	X	X	
	Remove from group			X
	Restore configuration	X		
	Update	X	X	
View Connection Server group	Set GUID of group			X
	Set name of group			X
View global setting	Get information	X	X	
	Update	X	X	
View service health monitor	Get information	X	X	X
Virtual machine (View Agent installed)	Get information	X	X	X
	Remove ownership	X	X	
	Reset	X	X	
	Update ownership	X	X	
VMware Horizon license	Get information	X	X	
	Set license	X	X	

View PowerCLI cmdlet Reference

You can use View PowerCLI cmdlets to administer View on a View Connection Server instance.

The following table lists all the available View PowerCLI cmdlets, organized by View object. For cmdlet syntax, use the `Get-Help` cmdlet. For more information, see [“Display Help for View PowerCLI,”](#) on page 28.

Table 3-4. View PowerCLI cmdlets

Object	cmdlet	Description
Desktop pool	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.
Linked-clone desktop pool	Add-AutomaticLinkedClonePool	Adds an automatically provisioned linked-clone desktop pool.
	Update-AutomaticLinkedClonePool	Updates an automatically provisioned linked-clone desktop pool.
	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.
	Export-NetworkLabelSpecForLinkedClone	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.
Full-clone desktop pool	Export-NetworkLabelSpecForFullClone	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.
Automatic virtual machine desktop pool	Add-AutomaticPool	Adds an automatically provisioned full virtual machine desktop pool.
	Update-AutomaticPool	Updates an automatically provisioned full virtual machine desktop pool.
Manual desktop pool	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.
View Composer domain	Get-ComposerDomain	Returns information about View Composer.
View Connection Server instance	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.
VMware Horizon license	Get-License	Returns the View licenses on a View Connection Server instance.
	Set-License	Sets a View license on a View Connection Server instance.

Table 3-4. View PowerCLI cmdlets (Continued)

Object	cmdlet	Description
Physical machine	Get-DesktopPhysicalMachine	Returns a list of physical machines that are available for use with unmanaged desktop pools.
Virtual machine	Get-DesktopVM	Returns information about virtual machines.
	Send-VMReset	Resets a virtual machine.
Event	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.
View global setting	Get-GlobalSetting	Returns global configuration information about the View environment.
	Update-GlobalSetting	Updates global configuration information about the View environment.
View service health monitor	Get-Monitor	Returns a list of health monitors for View services.
Persistent user data disk	Get-ProfileDisk	Returns information about persistent user data disks.
Remote session	Get-RemoteSession	Returns information about active remote sessions.
	Send-SessionDisconnect	Disconnects an active remote session.
	Send-SessionLogoff	Logs out an active remote session.
User	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.
vCenter Server instance	Add-ViewVC	Adds a vCenter Server instance to View.
	Get-ViewVC	Returns information about vCenter Server instances.
	Remove-ViewVC	Removes a vCenter Server instance from View.
	Update-ViewVC	Updates the configuration of a vCenter Server instance in View.

View PowerCLI cmdlet Parameters

Some View PowerCLI cmdlet parameters accept settings. For example, the `-flashQuality` parameter accepts settings that specify the maximum allowable quality for Adobe Flash content.

Default Display Protocol Parameter

The `-defaultProtocol` parameter specifies the default display protocol for a desktop pool.

Table 3-5. -defaultProtocol Parameter Settings

Setting	Description
PCOIP	Set the default display protocol to PCoIP.
RDP	Set the default display protocol to Microsoft RDP.

Deletion Policy Parameter

The `-deletePolicy` parameter specifies the deletion policy for automatically provisioned floating and linked-clone desktop pools.

Table 3-6. `-deletePolicy` Parameter Settings

Setting	Description
Default	Do not delete the machine when the user logs out.
DeleteOnUse	Delete the machine when the user logs out.
RefreshOnUse	Refresh the machine when the user logs out. NOTE This setting applies only to linked clone desktop pools.

Flash Quality Parameter

The `-flashQuality` parameter specifies the maximum allowable quality for Adobe Flash content. This value overrides the setting on a Web page. If the Adobe 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 Adobe Flash content causes the content to use less bandwidth.

Table 3-7. `-flashQuality` Parameter Settings

Setting	Description
HIGH	Allow low, medium, or high quality Flash content.
LOW	Allow only low quality Flash content.
MEDIUM	Allow low or medium quality Flash content.
NO_CONTROL	Allow the Web page settings to determine the quality of Flash content.

Flash Throttling Parameter

The `-flashThrottling` parameter specifies how often Adobe Flash refreshes onscreen information. Throttling Adobe Flash to increase the refresh interval reduces the frame rate. This reduction causes Adobe Flash content to use less bandwidth, but it might also cause Adobe Flash to drop frames.

Table 3-8. `-flashThrottling` Parameter Settings

Setting	Description
AGGRESSIVE	Set the refresh interval to 2500 milliseconds. This setting produces the highest number of dropped frames. The speed of audio transmission is unaffected.
CONSERVATIVE	Set the refresh interval to 100 milliseconds. This setting produces the lowest number of dropped frames. The speed of audio transmission is unaffected.
DISABLED	Disable throttling. The timer interval is not modified.
MODERATE	Set the refresh interval to 500 milliseconds. The speed of audio transmission is unaffected.

LDAP Backup Frequency Parameter

The `-ldapBackupFrequency` parameter specifies the LDAP backup frequency for a View Connection Server instance.

Table 3-9. -ldapBackupFrequency Parameter Settings

Setting	Description
Every12Hour	Back up the LDAP database once every 12 hours.
Every2Day	Back up the LDAP database once every two days.
Every2Week	Back up the LDAP database once every two weeks.
Every6Hour	Back up the LDAP database once every six hours.
EveryDay	Back up the LDAP database once per day.
EveryHour	Back up the LDAP database once per hour.
EveryWeek	Back up the LDAP database once per week.
Never	Turn off backup for the LDAP database.

Pool Type Parameter

The `-poolType` parameter specifies the desktop pool type.

Table 3-10. -poolType Parameter Settings

Setting	Description
IndividualUnmanaged	The pool contains an individual unmanaged machine.
IndividualVC	The pool contains an individual machine that is managed and configured by a vCenter Server instance.
Manual	The pool contains manually configured floating (nonpersistent) machines that are managed and configured by a vCenter Server instance.
ManualUnmanagedNonPersistent	The pool contains manually configured floating (nonpersistent) machines that are not managed by a vCenter Server instance.
ManualUnmanagedPersistent	The pool contains manually configured dedicated (persistent) machines that are not managed by a vCenter Server instance.
ManualVCPersistent	The pool contains manually configured dedicated (persistent) machines that are managed by a vCenter Server instance.
NonPersistent	(AutomaticPool) The pool contains automatically configured floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server instance.
OnRequestSviNonPersistent	(AutomaticPool) The pool contains floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server instance and View Composer when requested.
OnRequestSviPersistent	(AutomaticPool) The pool contains dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server instance and View Composer when requested.
OnRequestVcNonPersistent	(AutomaticPool) The pool contains floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server instance when requested.
OnRequestVcPersistent	(AutomaticPool) The pool contains dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server instance when requested.
Persistent	(AutomaticPool) The pool contains automatically configured dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server instance.

Table 3-10. -poolType Parameter Settings (Continued)

Setting	Description
SVINonPersistent	(AutomaticPool) The pool contains floating (nonpersistent) machines that are provisioned, managed, and configured by a vCenter Server instance and View Composer.
SVIPersistent	(AutomaticPool) The pool contains dedicated (persistent) machines that are provisioned, managed, and configured by a vCenter Server instance and View Composer.

Power Policy Parameter

The `-powerPolicy` parameter specifies the power policy for a desktop pool.

Table 3-11. -powerPolicy Settings

Setting	Description
AlwaysOn	Configure the machine to remain powered on, even when no one is using it. If you shut down the machine, the machine restarts immediately.
RemainOn	Start the machine when required if the machine is powered down. The machine remains powered on until you shut it down.
PowerOff	Shut down the machine when no one is using it.
Suspend	Suspend the machine when no one is using it.

Refresh Policy Type Parameter

The `-refreshPolicyType` parameter specifies the refresh policy for the OS disks of automatically provisioned dedicated and linked clone desktop pools.

Table 3-12. -refreshPolicyType Settings

Setting	Description
Always	Refresh the OS disk whenever the user logs off.
Conditional	Refresh the OS disk when the user logs off, but only if certain conditions are met. Use the <code>-refreshPolicyDays</code> and <code>-refreshPolicyUsage</code> parameters to specify the refresh interval in days and the percentage of the maximum allowable size for the disk.
Never	Never refresh the OS disk when the user logs off.

Smart Card Setting Parameter

The `-smartCardSetting` parameter specifies the smart card authentication policy setting. This setting applies to `Update-ConnectionBroker`, and does not have any effect unless you also specify `-UseSSLClient $true` with `Update-GlobalSetting`.

Table 3-13. -smartCardSetting Parameter Settings

Setting	Description
NotAllowed	Disable smart card authentication.
Optional	Allow users to 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	Require users to use smart card authentication when they connect to the View Connection Server instance. Smart card authentication replaces only Windows password authentication. If SecureID is enabled, users are required to authenticate by using both SecureID and smart card authentication.

View Composer Task Parameter

The -composerTask parameter specifies a View Composer maintenance task on a virtual machine.

Table 3-14. -composerTask Parameter Settings

Setting	Description
attachUdd	Attach a persistent disk.
detachUdd	Detach a persistent disk.
mkChkPoint	Create a checkpoint snapshot.
rebalance	Rebalance a linked clone machine.
replaceUdd	Replace a persistent disk.
resync	Recompose a linked clone machine.

Examples of Using View PowerCLI cmdlets

With View PowerCLI cmdlets, you can perform View management tasks from the command line or from scripts instead of using View Administrator.

Managing View Connection Server Instances

You can use View PowerCLI cmdlets to perform View Connection Server management tasks.

Table 3-15. Examples of Common View Connection Server Management Tasks

Task	Example View PowerCLI cmdlet Syntax
Get configuration settings for a specific View Connection Server instance	<code>Get-ConnectionBroker -broker_id CONNSVR1</code>
Update configuration settings for a specific View Connection Server instance	<code>Update-ConnectionBroker -broker_id CONNSVR1 -directConnect \$false -secureIdEnabled \$true -ldapBackupFrequency EveryWeek</code>
Configure secure PCoIP connections for a specific View Connection Server instance	<code>Update-ConnectionBroker -broker_id CS-VSG -directPCoIP \$FALSE</code>
Set the PCoIP external URL for a specific View Connection Server instance	<code>Update-ConnectionBroker -broker_id CS-VSG -externalPCoIPURL 10.18.133.34:4172</code>
Set the PCoIP external URL for a specific security server	<code>Update-ConnectionBroker -broker_id SECSVR-03 -externalPCoIPURL 10.116.32.136:4172</code>

Managing vCenter Server Instances in View

You can use View PowerCLI cmdlets to perform vCenter Server management tasks in View.

Table 3-16. Examples of Common vCenter Server Management Tasks in View

Task	Example View PowerCLI cmdlet Syntax
Add a vCenter Server instance to the View configuration	<code>Add-ViewVC -serverName vc01.mydom.int -username Administrator -password clydenw -createRampFactor 5 -deleteRampFactor 5</code>
Get information about a specific vCenter Server instance in View	<code>Get-ViewVC -serverName vc01.mydom.int</code>
Get information about all the vCenter Server instances in a specific DNS domain	<code>Get-ViewVC -serverName *.mycorp.com</code>
Change the ramp factor values for a specific vCenter Server instance	<code>Get-ViewVC -serverName svr11.mycorp.com Update-ViewVC -createRampFactor 5 -deleteRampFactor 10</code>
Change the create ramp factor value for all vCenter Server instances in a specific DNS domain	<code>Get-ViewVC -serverName *.mycorp.com Update-ViewVC -createRampFactor 5</code>
Remove a vCenter Server instance from the View configuration	<code>Get-ViewVC -serverName vc02.mydom.int Remove-ViewVC</code>

Managing Desktop Pools

You can use View PowerCLI cmdlets to perform desktop pool management tasks.

Table 3-17. Examples of Common Desktop Pool Management Tasks

Task	Example View PowerCLI cmdlet Syntax
Get information about a desktop pool that has a specific display name	<code>Get-Pool -displayName "My Pool 1"</code>
Get information about all desktop pools that have a display name with a specific prefix	<code>Get-Pool -pool_id mypool-*</code>
Get information about all desktop pools that are configured to use the PCoIP display protocol	<code>Get-Pool -protocol PCoIP</code>
Get information about all individual unmanaged desktop pools	<code>Get-Pool -poolType IndividualUnmanaged</code>
Remove a desktop pool that has a specific pool ID	<code>Remove-Pool -pool_id dtpool-10</code>
Remove a desktop pool that has a specific pool ID and terminate any active sessions, but do not delete its image from disk	<code>Remove-Pool -pool_id dtpool-12 -TerminateSession \$true -DeleteFromDisk \$false</code>
Remove a desktop pool that has a specific display name and delete its image from disk	<code>Get-Pool -displayName "My Pool 1" Remove-Pool -DeleteFromDisk \$true</code>

Creating and Updating Automatically Provisioned Desktop Pools

You can use the `Get-ViewVC` and `Update-AutomaticPool` cmdlets to create and update automatically provisioned desktop pools.

In the following example, the `Get-ViewVC` cmdlet adds an automatically provisioned desktop pool called `auto1`. The desktop pool is managed by a vCenter Server instance called `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
```

You can provision all the desktops in advance by setting the `-minimumCount` and `-maximumCount` parameters to the same value.

In the following example, the `Update-AutomaticPool` cmdlet updates the configuration of an automatically provisioned desktop pool called `auto1`.

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

Because the datastores specified in the `-dataStorePaths` parameter override the previous setting, you must specify any existing datastores in the parameter for the desktop pool to continue to use those datastores.

Creating and Updating Linked-Clone Desktop Pools

You can use the `Get-ViewVC` and `Get-DesktopVM` cmdlets to create and update linked-clone desktop pools.

In the following example, the `Get-ViewVC` cmdlet adds a linked-cloned desktop pool named `lcdpool_1`. The pool is managed by View Composer on a vCenter Server instance named `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
```

You can provision all the desktops in advance by setting the `-minimumCount` and `-maximumCount` parameters 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 a letter that conflicts with a network-mounted drive.

In the following example, the `Get-ViewVC` cmdlet updates the configuration of a linked-clone desktop pool named `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
```

Because the datastores specified in the `-dataStoreSpecs` parameter override the previous setting, you must specify any existing datastores in the parameter for the pool to continue to use those datastores.

You can use the `Get-DesktopVM` cmdlet to perform rebalance, refresh, and recompose operations.

Table 3-18. Examples of Rebalance, Refresh and Recompose Operations

Task	Example View PowerCLI cmdlet Syntax
Rebalance desktops among the available datastores in a linked-clone desktop pool	<code>Get-DesktopVM -pool_id lcdpool_2 Send-LinkedCloneRebalance -schedule 2011-05-10:01:00:00 -forceLogoff \$false -stopOnError \$true</code>
Refresh the operating system disk in each linked-clone desktop by restoring its original state and size	<code>Get-DesktopVM -pool_id lcdpool_2 Send-LinkedCloneRefresh -schedule "May 12 2011 01:15" -forceLogoff \$true -stopOnError \$true</code>
Recompose all linked-clone desktops from a snapshot of the parent virtual machine	<code>Get-DesktopVM -pool_id lcdpool_2 Send-LinkedCloneRecompose -schedule ((Get-Date).AddHours(8)) -parentVMPath /AutoPoolVMs/parent2 -parentSnapshotPath /AutoPoolSnapshots/parent2_snapshot -forceLogoff \$true -stopOnError \$true</code>

Creating and Updating Manually Provisioned Desktop Pools

You can use the `Add-ManualPool`, `Get-ViewVC`, and `Update-ManualPool` cmdlets to create and update manually provisioned desktop pools.

In the following example, the `Add-ManualPool` cmdlet creates a manually provisioned desktop pool named `manPool` that contains a virtual machine named `myVM`.

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

In the following example, the `Get-ViewVC` cmdlet creates a manually provisioned desktop pool named `man1` from the desktops managed by the vCenter Server instance named `vc.mydom.int`.

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

In the following example, the `Update-ManualPool` cmdlet updates the configuration of a manually provisioned desktop pool named `man1`.

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

NOTE To use the `Get-VM` cmdlet, you must install vSphere PowerCLI.

Creating and Updating Manual Unmanaged Desktop Pools

You can use the `Add-ManualUnmanagedPool` and `Update-ManualUnmanagedPool` cmdlets to create and update manual unmanaged desktop pools.

In the following example, the `Add-ManualUnmanagedPool` cmdlet creates an unmanaged desktop pool named `unman1` that contains physical machines named `pm01` and `pm02`.

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

In the following example, the `Update-ManualUnmanagedPool` cmdlet updates the configuration of an unmanaged desktop pool named `unman1`.

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

Displaying Information About Users and Groups

You can use the `Get-User` cmdlet to display information about Active Directory users and groups.

In the following example, the `Get-User` cmdlet displays information about all the users in a domain named `mydom`.

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

In the following example, the `Get-User` cmdlet displays information about a user named fred in the domain named mydom. It excludes information about the user's group.

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

Managing Desktop Entitlements

You can use View PowerCLI cmdlets to manage desktop entitlements.

Table 3-19. Examples of Common Desktop Entitlement Management Tasks

Task	Example View PowerCLI cmdlet Syntax
Entitle a user in a specific domain to a specific desktop pool	<code>Get-User -name "mydom\fred" Add-PoolEntitlement -pool_id dtop-12</code>
Entitle a user to all desktop pools	<code>Get-Pool Add-PoolEntitlement -sid (Get-User -name "usr1").sid</code>
Get information about all the users who are entitled to use a specific desktop pool	<code>Get-PoolEntitlement -pool_id dtop-1</code>
Get information about all the users who are entitled to use desktop pools that have IDs with a specific prefix	<code>Get-Pool -pool_id dtpool-* Get-PoolEntitlement</code>
Remove an entitlement to use a specific desktop pool	<code>Get-PoolEntitlement -pool_id dtpool-11 Remove-PoolEntitlement</code>
Remote all entitlements	<code>Get-PoolEntitlement Remove-PoolEntitlement -forceRemove \$true</code> NOTE If you do not specify the <code>-forceRemove</code> parameter, you can use this command to obtain information about the entitlements to be removed.

Managing Remote Sessions

You can use View PowerCLI cmdlets to manage remote sessions.

Table 3-20. Examples of Common Remote Session Management Tasks

Task	Example View PowerCLI cmdlet Syntax
List all active remote sessions for a specific domain user	<code>Get-RemoteSession -username mydom\fred</code>
Disconnect all active sessions for a specific domain user	<code>Get-RemoteSession -username mydom\fred Send-SessionDisconnect</code>
Log off all active remote sessions for a specific domain user	<code>Get-RemoteSession -username mydom\fred Send-SessionLogoff</code>
Log off all active remote sessions that use the RDP display protocol	<code>Get-RemoteSession -protocol RDP Send-SessionLogoff</code>

Managing Virtual Machines

You can use View PowerCLI cmdlets to manage virtual machines. View Agent must be running in the virtual machines.

Table 3-21. Examples of Common Virtual Machine Management Tasks

Task	Example View PowerCLI cmdlet Syntax
Get information about the virtual machines for a specific desktop pool	<code>Get-DesktopVM -pool_id dtpool-3</code>
Get information about the virtual machines configured on a specific vCenter Server instance	<code>Get-DesktopVM -vc_id (Get-ViewVC -serverName vc03.local.int).vc_id</code>
Get information about the virtual machines managed by the same vCenter Server instance that provisions a specific desktop pool	<code>Get-ViewVC -pool_id dtpool-1 Get-DesktopVM</code>
Get information about all the virtual machines managed by a specific vCenter Server instance	<code>Get-ViewVC -serverName vc01.mydom.int Get-DesktopVM</code>
List all the active persistent user data disks for a specific virtual machine	<code>Get-ProfileDisk -VMname vm01</code>
Reset all the virtual machines for a specific desktop pool	<code>Get-Pool -pool_id dtpool-05 Get-DesktopVM Send-VMReset</code>
Reset the virtual machine for a desktop pool that has a specific display name	<code>Get-Pool -displayName dtp1 Get-DesktopVM Send-VMReset</code>

Displaying Information About Physical Machines

You can use the `Get-DesktopPhysicalMachine` cmdlet to display information about a physical machine.

In this example, the `Get-DesktopPhysicalMachine` cmdlet displays information about a physical machine that has a specific IP address.

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

Updating Virtual Machine Ownership

You can use the `Update-UserOwnership` and `Remove-UserOwnership` cmdlets to update ownership for virtual machines.

In this example, the `Update-UserOwnership` cmdlet updates the ownership of a virtual machine named `vm04` for a user named `user1`.

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

In this example, the `Remove-UserOwnership` cmdlet removes the ownership of a virtual machine named `vm22`.

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

Displaying Event Reports

You can use View PowerCLI cmdlets to display event reports.

Table 3-22. Examples of Common Event Reporting Tasks

Task	Example View PowerCLI cmdlet Syntax
List all the available event reporting views	<code>Get-EventReportList</code>
Display all the configuration change events that occurred after a specific date	<code>Get-EventReport -viewName config_changes -startDate (Get-Date -Year 2011 -Month 5 -Day 20 -Hour 0 -Minute 0 -Second 0)</code>
Display all the user events that occurred between two specific dates	<code>Get-EventReport -viewName user_events -startDate (Get-Date -Year 2011 -Month 12 -Day 1 -Hour 0 -Minute 0 -Second 0) -endDate (Get-Date -Year 2011 -Month 12 -Day 2 -Hour 0 -Minute 0 -Second 0)</code>
Display all the user events that occurred during for the last 24 hours	<code>Get-EventReport -viewName user_events -startDate ((Get-Date).AddDays(-1))</code>
Display all the user events that occurred during the current year	<code>Get-EventReport -viewName user_events -startDate (Get-Date -Day 01 -Month 01 -Hour 0 -Minute 0 -Second 0)</code>

Displaying and Updating Global Settings

You can use View PowerCLI cmdlets to display and update global settings for View.

Table 3-23. Examples of Common Global Settings Management Tasks

Task	Example View PowerCLI cmdlet Syntax
Display the global settings	<code>Get-GlobalSetting</code>
Update the session timeout setting	<code>Update-GlobalSetting -SessionTimeout 1800</code>
Update the forced logout warning message and delay period	<code>Update-GlobalSetting -DisplayLogoffWarning \$true -ForcedLogoffAfter \$logoutdelay -ForcedLogoffMessage "Forced log out will occur in \$logoutdelay minutes"</code>
Require clients to use SSL to connect and set the prelogin message	<code>Update-GlobalSetting -UseSSLClient \$true -PreLoginMessage "Insert disclaimer and other notices here."</code>

Displaying and Adding License Keys

You can use the `Get-License` and `Set-License` cmdlets to display and add license keys for View.

In this example, the `Get-License` cmdlet displays the installed license keys.

```
Get-License
```

In this example, the `Set-License` cmdlet adds a license key.

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

Examples of Using View PowerCLI to Perform Advanced Tasks

You can combine View PowerCLI and vSphere PowerCLI cmdlets to create PowerShell functions that perform complex operations, such as resizing pools and adding datastores to desktop pools.

Determining if View Connection Server Is Running

The following PowerShell function determines whether the View Connection Server service is running and starts the service if it is not running.

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

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

```

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

```

Resizing Automatic and Linked-Clone Pools

The following PowerShell functions determine the current usage of all desktop pools and resize any automatically provisioned or linked-clone desktop pools that are at 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

The following PowerShell function uses vSphere PowerCLI to return the full path to a vSphere inventory object.

```

# 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

The following PowerShell function 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

You can define PowerShell functions to add and remove datastores.

The PowerShell functions in the following example add and remove a datastore for 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
}

```

The PowerShell functions in the following example add and remove a datastore for 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
}

```

Assigning Multiple Network Labels to a Desktop Pool

In View 5.2 and later releases, you can configure automated full-clone and linked-clone desktop pools to use multiple network labels. This feature expands the number of IP addresses you can assign to the virtual machines in a pool, making it easier to create pools that have a large number of desktops. You can use View PowerCLI cmdlets to assign network labels that are available in the vCenter Server resource pool where the automated full-clone or linked-clone desktop pool is deployed.

By default, the virtual machines in a desktop pool inherit the network interface card (NIC) and its associated network label from the parent virtual machine or template. Some parent virtual machines or templates might have multiple NICs and 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.

View distributes network labels among the virtual machines in the entire desktop pool. When View provisions desktops, it assigns network labels in alphabetical order. When the maximum number of virtual machines are provisioned with IP addresses that use the first network label, View starts to assign the second label, and so on.

To configure a desktop pool to use multiple network labels, you select network labels from the labels defined for the ESXi cluster in vCenter Server, associate the labels with the NICs that are inherited from a parent virtual machine or template, specify the maximum number of IP addresses that can be assigned to virtual machines from each network label, and save the information in a network label configuration file. You specify the network label configuration file in the View PowerCLI cmdlet that you use to create the desktop pool.

Network Label Configuration File Format

You use a network label configuration file to configure automated full-clone and linked-clone desktop pools to use multiple network labels. The network label configuration file contains a flag that controls whether network labels are assigned and sections that define NICs, network labels, and network label attributes.

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 Section

This section lists the NICs defined in vCenter Server on the template or snapshot of the parent virtual machine. Do not edit this section.

Parameter Definition for Network Section

This section lists the network labels defined in vCenter Server instance 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 Section

This section 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-NetworkLabelSpecForLinkedClone` or `Export-NetworkLabelSpecForFullClone` cmdlet. You can manually edit this value in the configuration file.

As a best practice, do not assign a network label to more than one desktop pool. The 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 or the IP assignments from the network label might become oversubscribed.

If the parent virtual machine or template has two NICs, each network label discovered by the `Export-NetworkLabelSpecForLinkedClone` or `Export-NetworkLabelSpecForFullClone` 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 aware only of its own network label assignments.

Example Network Label Configuration File

The following example shows a typical 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
```

Obtain and Export NIC and Network Label Information

Before you can assign multiple network labels to a pool, you must obtain NIC and network label information and export that information to a network label configuration file.

For a full-clone pool, you obtain information about the NICs configured on the template that you use to create the full-clone pool. For a linked-clone pool, you obtain information about the NICs configured on the parent virtual machine that you use to create the linked-clone pool. You obtain network label information from the available network labels configured on the ESXi hosts in the cluster on which the desktop pool is to be deployed.

When you export the NIC and network label information to a network label configuration file, you set the maximum number of virtual machines that can be provisioned with IP addresses from each network label. You should 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 number to 244.

Prerequisites

Become familiar with using View PowerCLI. See [“Getting Started with View PowerCLI,”](#) on page 27.

Procedure

- To obtain and export NIC and network label information to create a full-clone pool, use the `Export-NetworkLabelSpecForFullClone` cmdlet.

For example:

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

- To obtain and export NIC and network label information to create a linked-clone pool, use the `Export-NetworkLabelSpecForLinkedClone` cmdlet.

For example:

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

The network label cmdlet obtains the network labels configured in vCenter Server for all the ESXi hosts in the specified cluster, sets the maximum number of virtual machines that can be provisioned with IP addresses from each network label, and exports the list of NICs and network labels to the network label configuration file.

Example: Obtaining NIC and Network Label Information

In this example, the `Export-NetworkLabelSpecForLinkedClone` cmdlet obtains NICs from the Win7-Parent virtual machine and Snapshot1, and network label information from Cluster1. A maximum of 244 virtual machines can be provisioned with IP addresses from each network label. The information is exported to a network label configuration file named `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"
```

What to do next

Verify and edit the network label configuration file. See [“Verify and Edit a Network Label Configuration File,”](#) on page 53.

Verify and Edit a Network Label Configuration File

After you generate a network label configuration file, you must verify its contents and edit it to specify the NIC and network label pairings to assign to the pool.

Prerequisites

- Generate a network label configuration file. See [“Obtain and Export NIC and Network Label Information,”](#) on page 52.
- Become familiar with the network label configuration file format. See [“Network Label Configuration File Format,”](#) on page 50.

Procedure

- 1 In a text editor, open the network label configuration file.
- 2 For each NIC and network label pairing that you want to assign to the pool, remove the comments (### marks).
- 3 Verify that each network label is assigned to only one NIC.

- 4 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.

- 5 Save your changes.

Example: Edited Network Label Configuration File

In this example, network06 and network07 are assigned to nic1 (Network adapter 1) and network08 and network09 are assigned to nic2 (Network adapter 2). network01 through network05 remain commented and are not used. This configuration can 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
```

What to do next

Create a desktop pool that can use the multiple network labels. See [“Deploy a Desktop Pool That Uses Multiple Network Labels,”](#) on page 54.

Deploy a Desktop Pool That Uses Multiple Network Labels

You can use View PowerCLI cmdlets to deploy an automated full-clone or linked-clone desktop pool that uses multiple network labels.

Prerequisites

- Generate a network label configuration file. See [“Obtain and Export NIC and Network Label Information,”](#) on page 52.
- Verify and edit the network label configuration file. See [“Verify and Edit a Network Label Configuration File,”](#) on page 53.

Procedure

- ◆ Use a View PowerCLI cmdlet to deploy the desktop pool.

Option	Action
Deploy a linked-clone pool	Run the <code>Add-AutomaticLinkedClonePool</code> cmdlet and specify the path to the network label configuration file with the <code>-NetworkLabelConfigFile</code> parameter.
Deploy a full-clone desktop pool	Run the <code>Add-AutomaticPool</code> cmdlet and specify the path to the network label configuration file with the <code>-NetworkLabelConfigFile</code> parameter.

Example: Linked-Clone Desktop Pool Deployment

In this example, the `Add-AutomaticLinkedClonePool` cmdlet creates a linked-clone pool named `POOL_NAME` and assigns the network labels in the network label configuration file named `label.txt` file.

```
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"
```

Assigning Network Labels

As View provisions virtual machines in a pool, it assigns network labels 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 View assigns a network label to a virtual machine, it 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.

Preserving Network Labels During Recompose and Rebalance Operations

During View Composer recompose and rebalance operations, View attempts 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. You can configure a NIC with a standard network switch or distributed virtual network switch.

The following examples describe the rules that govern 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.

Displaying Network Label Assignments for a Pool

You can use the `Get-Pool` cmdlet to display the network label assignments for a pool. The `networkLabelSpecs` output parameter shows the network labels that are assigned to the pool. You do not need to add a parameter in the command line to display this information.

The following table describes the `networkLabelSpecs` output format.

Table 3-24. `networkLabelSpecs` Output Format

Value	Description
<code>nl</code>	Network label name.
<code>nic</code>	NIC name.
<code>enabled</code>	A value of 1 means the label is active. A value of 0 means it is disabled.
<code>max</code>	Maximum number of virtual machines that can be assigned the network label.
<code>usage</code>	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 `Get-Pool` cmdlet example shows network label assignments for the pool `Pool2`.

```
> get-pool -pool_id Pool2
...
networkLabelSpecs : [nl=desktop-auto01-230;nic=Network adapter 1;enabled=1;max=239;usage=239];[nl=desktop-auto02-240;nic=Network adapter 1;enabled=1;max=239;usage=239];[nl=desktop-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;usage=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;enabled=1;max=239;usage=239];[nl=desktop-auto09-310;nic=Network adapter 1;enabled=1;max=239;usage=88] ...
```

Displaying Network Label Assignments for a Virtual Machine

You can use the `Get-DesktopVM` cmdlet to display network label assignments for a virtual machine. The `netLabelAutoAssigns` output parameter shows the NIC names and network labels that View attempted to assign to the virtual machine. You do not have to add a parameter in the command line to display this information.

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

The `netLabelAutoAssigns` output displays the network label assignments that View Connection Server reserves for the virtual machine. To see the network labels that are assigned to the virtual machine in vCenter Server, use the `-getNetworkLabel` parameter with the `Get-DesktopVM` cmdlet. For more information, see [“Displaying vCenter Server Network Label Assignments for a Virtual Machine,”](#) on page 57.

The following `Get-DesktopVM` cmdlet 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 vCenter Server Network Label Assignments for a Virtual Machine

vCenter Server can make additional network label assignments to virtual machines. These assignments are outside the control of View.

To see the network labels that are assigned to a virtual machine in vCenter Server, use the `-getNetworkLabel` parameter with the `Get-DesktopVM` cmdlet. You must type a Boolean value of `$true` in the command line to enable the `-getNetworkLabel` parameter. 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.

The following `Get-DesktopVM` cmdlet 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} ...

```

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

Disable Automatic Network Label Assignments

You can disable network label assignments on an existing pool that uses automatic assignments.

Procedure

- 1 In a text editor, open the network label configuration file.
- 2 Set the enabled flag to false.
For example: `enabled=false`
- 3 Save your changes.
- 4 Run the `Update-AutomaticLinkedClonePool` or `Update-AutomaticPool` cmdlet and specify the path to the updated network label configuration file with the `-NetworkLabelConfigFile` parameter.

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

Customizing LDAP Data

You can use VMware and Microsoft command-line tools to import and export LDAP configuration data to and from View. These command-line tools import and export LDAP configuration data in LDAP Data Interchange Format (LDIF) configuration files.

This feature is intended for use by advanced administrators who want to perform automatic bulk configuration operations. To create scripts to update the View configuration, use View PowerCLI.

This chapter includes the following topics:

- [“Introduction to LDAP Configuration Data,”](#) on page 59
- [“Modifying LDAP Configuration Data,”](#) on page 60

Introduction to 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 other repositories.

When you use View Administrator to modify the View configuration, the appropriate LDAP data is updated in the repository. For example, if you add a desktop pool, View stores information about users, user groups, and entitlements in LDAP. 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 configuration files to perform a number of tasks, including transferring configuration data between View Connection Server instances and backing up your View configuration so that you can restore the state of a View Connection Server instance.

You can also use LDIF configuration files to define a large number of View objects, such as desktop pools, and add those objects to your View Connection Server instances without having to use View Administrator to perform the task manually.

In View 3.1 and later releases, View performs regular backups of the LDAP repository.

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

Modifying LDAP Configuration Data

You can export LDAP configuration data on a View Connection Server instance to an LDIF configuration file, modify the LDIF configuration file, and import the modified LDIF configuration file into other View Connection Server instances to perform automatic bulk configuration operations.

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

Export LDAP Configuration Data

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

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

Procedure

- 1 Log in to a standard or replica View Connection server instance.

Option	Action
View 3.1 and earlier	Log in as an administrator and be a member of the Local Administrators user group.
View 4.5 and later	Log in as a user in the Administrators or Administrators (Read only) role. NOTE You must be logged in as a user in the Administrators or Administrators (Read only) role to export configuration data from the View configuration repository.

- 2 At the command prompt, type the `vdmexport` command and use the `-f` option to specify the name of the LDIF configuration file to export.

For example: `vdmexport -f myexport.LDF`

Alternatively, you can redirect the output instead of using the `-f` option.

For example: `vdmexport > myexport.LDF`

The `vdmexport` 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.

Defining a Desktop Pool in an LDIF Configuration File

You can define a desktop pool in an LDIF configuration file and import the customized LDIF configuration file to create a large number of desktop pools.

NOTE You can also create customized LDIF configuration files for other objects that are defined in the LDAP repository, including global configuration settings, configuration settings for a specific View Connection Server instance or security server, and configuration settings for a specific user.

To define a desktop pool in an LDIF configuration file, you must add the following entries to the file.

- 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

You associate each VM Pool entry 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 table describes the attributes you must specify when you modify a desktop pool definition in an LDIF configuration file.

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

Entry	Attribute	Description
Virtual Desktop VM VM Pool Desktop Application	cn	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 <code>System.Guid.NewGuid().ToString()</code> in Visual Basic).
Desktop Application	member	<p>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 <code><SID=S-1-2-3-4></code> represents an AD user or group with the SID value S-1-2-3-4.</p> <p>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, <code>PFNJRD1TLTEtMi0zLTQ+IA==</code>).</p> <p>Because this attribute is multivalued, you can use it on multiple lines to represent each entry in a list of SIDs.</p>

Sample LDIF Configuration File Desktop Pool Entries

The following example is an excerpt from an LDIF configuration file. It 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

```

Use the vdmimport Command to Import LDAP Configuration Data

In View 4.5 and later releases, you can use the `vdmimport` command to import configuration data from an LDIF configuration file into a standard or replica View Connection Server instance.

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

Prerequisites

- Verify that you have View 4.5 or later. If you have an earlier View release, see [“Use the LDIFDE Command to Import LDAP Configuration Data,”](#) on page 64.
- Export LDAP configuration data to an LDIF configuration file. See [“Export LDAP Configuration Data,”](#) on page 60.

Procedure

- 1 Log in to a View Connection Server instance as a user in the Administrators role.

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

- At the command prompt, type the `vdmimport` command and use the `-f` option to specify the LDIF configuration file to import.

For example: `vdmimport -f myexport.LDF`

After the `vdmimport` command runs, 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.

Use the LDIFDE Command to Import LDAP Configuration Data

You can use the Microsoft LDIFDE command to import configuration data from an LDIF configuration file into a standard or replica View Connection Server instance.

In View releases earlier than View 4.5, you must use the Microsoft LDIFDE command to import configuration data from an LDIF configuration file. The `vdmimport` command is not supported in View releases earlier than View 4.5.

If you have View 4.5 or later, use the `vdmimport` command rather than the LDIFDE command. The `vdmimport` command does not display the large number of error messages that are produced by running the LDIFDE command. For more information, see [“Use the vdmimport Command to Import LDAP Configuration Data,”](#) on page 63.

Because the LDIFDE command does not update, create, or delete any LDAP records that are not defined in the LDIF configuration file, it enables you to customize an LDIF configuration file so that only selected records are affected when you import the file. For complete information about using the LDIFDE command, go to <http://support.microsoft.com/kb/237677>.

Prerequisites

Export LDAP configuration data to an LDIF configuration file. See [“Export LDAP Configuration Data,”](#) on page 60.

Procedure

- Log in to a View Connection server instance.

Option	Action
View 3.1 or earlier	Log in as an administrator and be a member of the Local Administrators user group.
View 4.5 or later	Log in as a user in the Administrators role. NOTE You must be logged in as a user in the Administrators role to import configuration data into the View configuration repository.

- At the command prompt, type the LDIFDE command and use the `-f` option to specify an existing LDIF configuration file.

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

After the LDIFDE command runs, the configuration of your View Connection Server instance is updated with the data from the file, and the number of records that are successfully updated is displayed.

Error messages appear whenever an existing entry in the repository is overwritten. You can ignore these error messages. Error messages also appear if a record cannot be updated because your role has insufficient privileges.

Integrating View with Microsoft SCOM

5

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

This chapter includes the following topics:

- [“Setting Up a SCOM Integration,”](#) on page 65
- [“Monitoring View in the Operations Manager Console,”](#) on page 70

Setting Up a SCOM Integration

Integrating View with SCOM involves assigning a name to the View Connection Server group, importing the View management packs on the SCOM server, enabling a proxy agent on each View Connection Server instance and security server, and running the View discovery script in the Operations Manager console.

Assign a Name to the View Connection Server Group

Before you can use SCOM to monitor and manage the state of View components, you must assign a name to the View Connection Server group in View. The Operations Manager console displays this name to help you identify the View Connection Server group within SCOM.

NOTE View Administrator does not display the View Connection Server group name.

Prerequisites

Become familiar with the `vdadmin` command-line interface. For more information, see the *View Administration* document.

Procedure

- 1 Log in to one of the View Connection Server hosts in the View Connection Server group.
- 2 At the command prompt, type the `vdadmin` command with the `-C` and `-c` options.

For example: `vdadmin -C -c group_name`

The `-c` option specifies the name to assign to the View Connection Server group.

Example: Assigning a View Connection Server Group Name

In this example, the `vdadmin` command sets the name of a View Connection Server group to `VCSG01`.

```
vdadmin -C -c VCSG01
```

What to do next

Complete the procedure described in “[Import the View Management Packs on the SCOM Server](#),” on page 66.

View Management Packs

View management packs enable you to use SCOM to monitor and manage the state of View components.

Table 5-1. View Management Packs

View Management Pack	Description
VMware.View.Discovery.mp	Contains the agent that discovers instances of View Server installations.
VMware.View.Monitoring.mp	Contains the views and monitors that you can use with View in the Operations Manager console.
VMware.View.Library.mp	Contains class and relationship definitions for the managed objects in View.
VMware.View.Image.Library.mp	Contains graphics that represent the classes defined in VMware.View.Library.mp.

The View management packs are installed in the C:\Program Files\VMware\VMware View\Server\extras\ManagementPacks directory on a View Connection Server instance or security server when you install the View Connection Server software.

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

Import the View Management Packs on the SCOM Server

You must import the View management packs on the SCOM server to use SCOM to monitor and manage the state of View components.

IMPORTANT McAfee VirusScan Enterprise 8.0i blocks the operation of Visual Basic scripts that SCOM uses. For more information and details about the available patch, go to <http://support.microsoft.com/kb/890736/en-us>.

Prerequisites

- Complete the procedure described in “[Assign a Name to the View Connection Server Group](#),” on page 65.
- Become familiar with the View management packs. See “[View Management Packs](#),” on page 66.

Procedure

- 1 Copy the View management packs from the View Connection Server instance or security server to the SCOM server.

The View management packs are in the C:\Program Files\VMware\VMware View\Server\extras\ManagementPacks directory on the View Connection Server host or security server.

- 2 In the Operations Manager console, go to **Administration\Management Packs** and select **Import Management Packs**.
- 3 Use the Import Management Packs wizard to import the View management packs.

What to do next

Complete the procedure described in “[Enable a Proxy Agent on a View Connection Server Host or Security Server](#),” on page 67.

Enable a Proxy Agent on a View Connection Server Host or Security Server

You must use the Operations Manager console to enable a proxy agent on each View Connection Server host or security server that you want to monitor with SCOM. The discovery script can discover a Windows server only if you enable the proxy agent for the server.

Prerequisites

Complete the procedure described in [“Import the View Management Packs on the SCOM Server,”](#) on page 66.

Procedure

- 1 In the Operations Manager console, go to **Administration\Agent Managed**, select the server, and click **Properties**.
- 2 On the **Security** tab, select the **Allow this agent to act as a proxy and discover managed objects on other computers** option.
- 3 Click **OK** to save your changes.

Run the Discovery Script in the Operations Manager Console

The discovery script finds systems on which a View server is installed. It 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.

NOTE Running the discovery script manually is optional. The discovery script is scheduled to run automatically once every hour.

Prerequisites

- Complete the procedure described in [“Import the View Management Packs on the SCOM Server,”](#) on page 66.
- Complete the procedure described in [“Enable a Proxy Agent on a View Connection Server Host or Security Server,”](#) on page 67. The discovery script can discover a Windows server only if you use the Operations Manager console to enable the proxy agent for the server.

Procedure

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

If the discovery script detects that a View server is installed on a computer, it creates instances of the View object classes that are defined in the `VMware.View.Library` management pack and establishes relationships between these managed objects.

For a list of the managed objects for View Connection Server instances and security servers, see [“View Connection Server and Security Server Managed Objects,”](#) on page 68. For information about the View object classes and their relationships, see [“View Object Classes and Relationships,”](#) on page 68.

What to do next

(Optional) Verify the objects that the discovery script creates for a server by viewing the objects in the Operations Manager console. See [“Display Discovered and Managed View Objects,”](#) on page 72.

View Connection Server and Security Server Managed Objects

The discovery script discovers managed objects for View Connection Server instances and security servers.

Table 5-2. Managed Objects for View Connection Server Instances and Security Servers

Object	View Connection Server Instance	Security Server
VMware.View.Cluster	X	X
VMware.View.Cluster.Node.Item	X	X
VMware.View.ConnectionServerRole.Item	X	X
VMware.View.Component.ConnectionServer.Item	X	X
VMware.View.Component.Framework.Item	X	X
VMware.View.Component.Web.Item	X	
VMware.View.Component.Directory.Item	X	
VMware.View.Component.SecureGateway.Item	X	X
VMware.View.Component.MessageBus.Item	X	
VMware.View.Component.SecurityServer.Item		X

View Object Classes and Relationships

The VMware.View.Library management pack contains class and relationship definitions for the View management packs. A class can have properties, such as a name or 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.

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

View Connection Server Group Classes

The VMware.View.Library management pack contains View Connection Server group class definitions.

Table 5-3. 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.

Base Classes

The VMware.View.Library management pack contains abstract base class definitions.

NOTE The currently supported instances that are derived from these classes must be View 5.1.x or a later release.

Table 5-4. View Library Base Classes

Class Name	Description
<code>VMware.View.Cluster.Node</code>	Represents a member of a View Connection Server group. This class has the properties <code>ClusterID</code> , <code>ClusterName</code> , <code>ProductVersion</code> , and <code>InstallPath</code> .
<code>VMware.View.Component</code>	Represents a View component that has been installed on a member of a View Connection Server group. This class has the property <code>Name</code> .
<code>VMware.View.Component.ConnectionServer</code>	Represents the Connection Server component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component</code> .
<code>VMware.View.Component.Directory</code>	Represents the Directory component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component</code> .
<code>VMware.View.Component.Framework</code>	Represents the Framework component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component</code> .
<code>VMware.View.Component.MessageBus</code>	Represents the Message Bus component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component</code> .
<code>VMware.View.Component.SecurityGateway</code>	Represents the Security Gateway component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component</code> .
<code>VMware.View.Component.SecurityServer</code>	Represents the Security Server component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component</code> .
<code>VMware.View.Component.Web</code>	Represents the Web component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component</code> .
<code>VMware.View.ConnectionServerRole</code>	Represents a member of a View Connection Server group with the Connection Server installed on it. This class inherits its properties from <code>VMware.View.NodeRole</code> .
<code>VMware.View.NodeRole</code>	Represents the role of a member of a View Connection Server group.
<code>VMware.View.SecurityServerRole</code>	Represents a member of a View Connection Server group with the Security Server installed on it. This class inherits its properties from <code>VMware.View.NodeRole</code> .

Concrete Classes

The `VMware.View.Library` management pack contains concrete class definitions.

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

Table 5-5. View Library Concrete Classes

Class Name	Description
<code>VMware.View.Cluster.Node.Item</code>	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 <code>VMware.View.Cluster.Node</code> .
<code>VMware.View.Component.ConnectionServer.Item</code>	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 <code>VMware.View.Component.ConnectionServer</code> .

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

Class Name	Description
<code>VMware.View.Component.Directory.Item</code>	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 <code>VMware.View.Component.Directory</code> .
<code>VMware.View.Component.Framework.Item</code>	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 <code>VMware.View.Component.Framework</code> .
<code>VMware.View.Component.MessageBus.Item</code>	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 <code>VMware.View.Component.MessageBus</code> .
<code>VMware.View.Component.SecurityGateway.Item</code>	Gateway component that has been installed on a member of a View Connection Server group. This class inherits its properties from <code>VMware.View.Component.SecureGateway</code> .
<code>VMware.View.Component.SecurityServer.Item</code>	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 <code>VMware.View.Component.SecurityServer</code> .
<code>VMware.View.Component.Web.Item</code>	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 <code>VMware.View.Component.Web</code> .
<code>VMware.View.ConnectionServerRole.Item</code>	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 <code>VMware.View.NodeRole</code> .
<code>VMware.View.SecurityServerRole.Item</code>	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 <code>VMware.View.NodeRole</code> .

Monitoring View in the Operations Manager Console

When View is integrated with SCOM, you can use the Operations Manager console to monitor and manage View components.

Views and Monitors to Use with View

The `VMware.View.Monitoring` management pack contains the views and monitors that you can use to monitor and manage View components in the Operations Manager console.

Views Available for View Objects

You can use the views defined in the `VMware.View.Monitoring` management pack to examine discovered View objects.

Table 5-6. Available Views for View Objects

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.

Table 5-6. Available Views for View Objects (Continued)

View	Description
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. <ul style="list-style-type: none"> ■ All Sessions ■ All Sessions High ■ SVI Sessions ■ SVI Sessions High
Secure Gateway Role Performance Data	Displays the following data sets. <ul style="list-style-type: none"> ■ Secure Gateway Sessions ■ Secure Gateway Sessions High

Available Monitor Types for View Objects

The VMware.View.Monitoring management pack 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.
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.
Domain connectivity monitor	Verifies that a View Connection Server instance can bind to all the domains of which it is a member. The monitor queries the status of the Web component on a View Connection Server instance every three minutes. If a View Connection Server instance cannot bind to a domain, SCOM sets its state to error and raises an alert.
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 three 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 Server instances. The monitor queries the Web component every three minutes for this information and raises an alert if a vCenter Server instance is not available.

Service Component Monitors for View Connection Server Instances

The following table describes the service component monitors that the VMware.View.Monitoring management pack provides for View Connection Server instances.

Table 5-7. 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 5-7. 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

Service Component Monitors for Security Servers

The following table describes the service component monitors that the `VMware.View.Monitoring` management pack provides for security servers.

Table 5-8. View Server 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

Display Discovered and Managed View Objects

You can display discovered and managed View objects in the Operations Manager console.

Prerequisites

Integrate View with SCOM. See [“Setting Up a SCOM Integration,”](#) on page 65.

Procedure

- To display the View objects that the discovery script creates for a server, go to **Monitoring\Discovery Inventory** in the Operations Manager console.
- To display the View objects that SCOM manages and the relationships between those objects, go to **Monitoring\VMware View** in the Operations Manager console and select the required view.

Display Performance Information

You can display graphical performance data for a View Connection Server instance or security server in the Operations Manager console.

Prerequisites

Integrate View with SCOM. See [“Setting Up a SCOM Integration,”](#) on page 65.

Procedure

- 1 In the Operations Manager console, go to **Monitoring\VMware View\Performance**.
- 2 Select the **Connection Server Role Performance Data** or **Secure Gateway Role Performance Data** view.
- 3 Select the required data sets.

Display Alerts for a View Connection Server Group

You can use the Health Explorer in the Operations Manager console to display information about alerts that the health monitors raise for a View Connection Server group.

Prerequisites

Integrate View with SCOM. See [“Setting Up a SCOM Integration,”](#) on page 65.

Procedure

- 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.

Close an Alert

You can close an alert in the Operations Manager console without taking any action.

NOTE This method does not prevent the alert from being raised again if the underlying cause persists.

Prerequisites

Integrate View with SCOM. See [“Setting Up a SCOM Integration,”](#) on page 65.

Procedure

- 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.

Restart a View Component Service

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

Prerequisites

Integrate View with SCOM. See [“Setting Up a SCOM Integration,”](#) on page 65.

Procedure

- 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 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 connectivity monitoring.

Prerequisites

Integrate View with SCOM. See [“Setting Up a SCOM Integration,”](#) on page 65.

Procedure

- 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 the View Connection Server instance and select **Open > Health Explorer**.
- 3 Right-click the **Domain Connectivity Health** entry for the View Connection Server instance in the Health Explorer and select **Monitor Properties**.
- 4 On 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, type the name of the excluded domain in the **Override Setting** text box, and select the **Enforced** check box.

To exclude more than one domain, use spaces to separate the domain names.

- 6 Click **Apply** and then click **OK** to save your changes.

Examining PCoIP Session Statistics with WMI

6

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&dis playlang=en>.

This chapter includes the following topics:

- [“Using PCoIP Session Statistics,”](#) on page 75
- [“General PCoIP Session Statistics,”](#) on page 76
- [“PCoIP Audio Statistics,”](#) on page 76
- [“PCoIP Imaging Statistics,”](#) on page 77
- [“PCoIP Network Statistics,”](#) on page 78
- [“PCoIP USB Statistics,”](#) on page 79
- [“Examples of Using PowerShell cmdlets to Examine PCoIP Statistics,”](#) on page 80

Using PCoIP Session Statistics

The WMI namespace for the PCoIP session statistics is `root\CIMV2`. The names of the statistics are suffixed with (Server) or (Client), according to whether the statistic is recorded on the PCoIP server or PCoIP client.

You can use Windows Performance Monitor (PerfMon) with the counters to calculate averages over a specified sampling period. You must have administrator privileges to access the performance counters remotely.

All statistics are reset to 0 when a PCoIP session is closed. If the WMI `SessionDurationSeconds` property 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 PCoIP Session Statistics

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

Table 6-1. General Session 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.

Calculating Bandwidth for Received PCoIP Data

To calculate the bandwidth in kilobytes per second for received PCoIP data over the time interval from time t1 to time t2, use the following formula.

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

Calculating Bandwidth for Transmitted PCoIP Data

To calculate the bandwidth in kilobits per second for transmitted PCoIP data over the time interval from time t1 to time t2, use the following formula.

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

Calculating Packet Loss for Received PCoIP Data

To calculate the percentage of received packets that are lost, use the following formula.

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

Calculating Packet Loss for Transmitted PCoIP Data

To calculate the percentage of transmitted packets that are lost, use the following formula.

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

PCoIP Audio Statistics

The WMI class name for PCoIP audio statistics is
Win32_PerfRawData_TeradiciPerf_PCoIPSessionAudioStatistics.

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

Table 6-2. PCoIP 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.
AudioRXBkbitPersec	Bandwidth for ingoing audio packets averaged over the sampling period, in seconds.
AudioTXBkbitPersec	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.

Calculating Bandwidth for Received Audio Data

To calculate the bandwidth in kilobits per second for received audio data over the time interval from time t1 to time t2, use the following formula.

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

Do not use AudioTXBkbitPersec for this calculation.

Calculating Bandwidth for Transmitted Audio Data

To calculate the bandwidth in kilobits per second for transmitted audio data over the time interval from time t1 to time t2, use the following formula.

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

Do not use AudioTXBkbitPersec for this calculation.

PCoIP Imaging Statistics

The WMI class name for PCoIP imaging statistics is Win32_PerfRawData_TeradiciPerf_PCoIPSessionImagingStatistics.

Table 6-3. PCoIP 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.
ImagingRXBkbitPersec	Bandwidth for incoming imaging packets averaged over the sampling period, in seconds.
ImagingTXBkbitPersec	Bandwidth for outgoing imaging packets averaged over the sampling period, in seconds.

Calculating Bandwidth for Received Imaging Data

To calculate the bandwidth in kilobits per second for received imaging data over the time interval from time t1 to time t2, use the following formula.

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

Do not use ImagingRXBkbitPersec for the calculation.

Calculating Bandwidth for Transmitted Imaging Data

To calculate the bandwidth in kilobits per second for transmitted imaging data over the time interval from time t1 to time t2, use the following formula.

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

Do not use ImagingTXBkbitPersec for the calculation.

PCoIP Network Statistics

The WMI class name for PCoIP network statistics is Win32_PerfRawData_TeradiciPerf_PCoIPSessionNetworkStatistics.

Table 6-4. PCoIP Network Statistics

WMI Property Name	Description
RoundTripLatencymsec	Round trip latency in milliseconds between the PCoIP server and the PCoIP client.
RXBkbitPersec	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.
TXBkbitPersec	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. <ul style="list-style-type: none"> ■ GPO bandwidth limit for the PCoIP client ■ GPO bandwidth limit for the PCoIP server ■ Bandwidth limit for the local network connection ■ Negotiated bandwidth limit for the Zero Client firmware based on encryption limits
TXPacketLossPercent	Percentage of transmitted packets lost during a sampling period.

Calculating Bandwidth for Received Network Data

To calculate the bandwidth in kilobits per second for received data over the time interval from time t1 to time t2, use the following formula.

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

Do not use RXBkbitPersec for the calculation.

Calculating Bandwidth for Transmitted Network Data

To calculate the bandwidth in kilobits per second for transmitted data over the time interval from time t1 to time t2, use the following formula.

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

Do not use TXBWkbitPersec for the calculation.

Calculating Packet Loss for Received Network Data

To calculate the packet loss in percentage for received data over the time interval from time t1 to time t2, use the following formula.

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

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

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

Do not use RXPacketLostPercent or RXPacketLostPercent_Base for the calculation.

Calculating Packet Loss for Transmitted Network Data

To calculate the packet loss in percentage for transmitted data over the time interval from time t1 to time t2, use the following formula.

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

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

$$\text{TXPacketsLost \%} = \text{TXPacketsLost during interval} / (\text{TXPacketsLost during interval} + \text{PacketsSent during interval}) * 100$$

Do not use TXPacketLostPercent or TXPacketLostPercent_Base for the calculation.

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

PCoIP USB Statistics

The WMI class name for PCoIP USB statistics is Win32_PerfRawData_TeradiciPerf_PCoIPSessionUSBStatistics.

Table 6-5. PCoIP 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.

Calculating Bandwidth for Received USB Data

To calculate the bandwidth in kilobits per second for received USB data over the time interval from time t_1 to time t_2 , use the following formula.

$$(\text{USBBytesReceived}[t_2] - \text{USBBytesReceived}[t_1]) * 8 / (1024 * (t_2 - t_1))$$

Do not use `USBRXBWkbitPersec` for the calculation.

Calculating Bandwidth for Transmitted USB Data

To calculate the bandwidth in kilobits per second for transmitted USB data over the time interval from time t_1 to time t_2 , use the following formula.

$$(\text{USBBytesSent}[t_2] - \text{USBBytesSent}[t_1]) * 8 / (1024 * (t_2 - t_1))$$

Do not use `USBTXBWkbitPersec` for the calculation.

Examples of Using PowerShell cmdlets to Examine PCoIP Statistics

You can use PowerShell cmdlets to examine PCoIP statistics.

In the following example, the `Get-WmiObject` cmdlet retrieves the PCoIP network statistics for the client `cm-02`.

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

In the following example, the `Get-WmiObject` cmdlet retrieves 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"
```

Setting Desktop Policies with Start Session Scripts

7

With start session scripts, you can configure specific View desktop settings before a desktop session starts based on information received from Horizon Client and View Connection Server.

For example, you can use a start session script to configure desktop policies based on client device and user location instead of setting up multiple desktop pools that have different desktop policies. A start session script can enable mapped drives, clipboard redirection, and other desktop features for a user who has an IP address in your organization's internal domain, but disallow these features for a user who has an IP address in an external domain.

This chapter includes the following topics:

- [“Obtaining Input Data for a Start Session Script,”](#) on page 81
- [“Best Practices for Using Start Session Scripts,”](#) on page 81
- [“Preparing a View Desktop to Use a Start Session Script,”](#) on page 82
- [“Sample Start Session Scripts,”](#) on page 84

Obtaining Input Data for a Start Session Script

Start session scripts cannot run interactively. A start session script runs in an environment created by View and must obtain its input data from that environment.

Start session scripts gather input data from environment variables on the client computer. Start session environment variables have the prefix `VDM_StartSession_`. For example, the start session environment variable that contains the client system's IP address is `VDM_StartSession_IP_Address`. You must ensure that a start session script validates the existence of any environment variable that it uses.

For a list of variables similar to start session environment variables, see “Client System Information Sent to View Desktops” in the *Setting Up Desktop and Application Pools in View* document.

Best Practices for Using Start Session Scripts

Follow these best practices when using start session scripts.

When to Use Start Session Scripts

Use start session scripts only if you need to configure desktop policies before a session starts.

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 session, rather than using start session scripts, satisfies most use cases.

For more information, see “Running Commands on View Desktops” in the *Setting Up Desktop and Application Pools in View* document.

Managing Start Session Timeouts

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 that View Connection Server sends. A long-running script is likely to cause the `StartSession` request to time out.

If a timeout occurs and the pool uses floating assignments, View Connection Server tries to connect the user to another virtual machine. If a timeout occurs and no virtual machine is available, View Connection Server rejects the user's connection request.

As a best practice, set a hard timeout for the script host operation so that a specific error can be returned if a script runs too long.

Making Start Session Scripts Accessible

The path where you configure your start session scripts must be 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 start session scripts in the `View_Agent_install_path\scripts` directory, for example:

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

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

Preparing a View Desktop to Use a Start Session Script

To prepare a View desktop to use a start session script, you must enable the VMware View Script Host service and add entries in the Windows registry.

You must configure all View desktops that need to run start session scripts. View does not provide a mechanism to propagate registry changes, VMware View Script Host service configuration changes, and start session scripts to multiple View desktop virtual machines.

Enable the VMware View Script Host Service

You must enable the VMware View Script Host service on each View desktop virtual machine where you want View to run a start session script. The VMware View Script Host service is disabled by default.

When you configure the VMware View Script Host service, you can optionally specify the user account under which the start session script runs. Start session scripts run in the context of the VMware View Script Host service. By default, the VMware View Host Script service is configured to run as the SYSTEM user.

IMPORTANT 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.

Procedure

- 1 Log in to the View desktop virtual machine.
- 2 At the command prompt, type `services.msc` to start the Windows Services tool.
- 3 In the details pane, right-click the VMware View Script Host service entry and select **Properties**.
- 4 On the **General** tab, select **Automatic** from the **Startup type** drop-down menu.
- 5 (Optional) If you do not want the local System account to run the start session script, select the **Log On** tab, select **This account**, and type the user name and password of the account to run the start session script.
- 6 Click **OK** and exit the Windows Services tool.

Add Windows Registry Entries for a Start Session Script

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

Prerequisites

- Verify that the path where you configured your start session scripts is accessible only to the SYSTEM account and local administrators. For more information, see [“Making Start Session Scripts Accessible,”](#) on page 82.
- 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 that View Connection Server sends. For more information, see [“Managing Start Session Timeouts,”](#) on page 82.

Procedure

- 1 Log in to the View desktop virtual machine.
- 2 At the command prompt, type `regedit` to start the Windows Registry Editor.
- 3 In the registry, navigate to `HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents`.
- 4 Add the path to the start session script to the registry.
 - a In the navigation area, right-click `ScriptEvents`, select **New > Key**, and create a key named `StartSession`.
 - b In the navigation area, right-click `StartSession`, select **New > String Value**, and create a string value that identifies the start session script to 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 particular order, invoke them from a single control script.
 - c In the topic area, right-click the entry for the new string value and select **Modify**.
 - d In the **Value data** text box, type the command line that invokes the start session script and click **OK**.

Type the full path of the start session script and any files that it requires.
- 5 Add and enable a start session value in the registry.
 - a Navigate to `HKLM\SOFTWARE\VMware, Inc.\VMware VDM\Agent\Configuration`.
 - b (Optional) If the `Configuration` key does not exist, right-click **Agent**, select **New > Key**, and create the key.
 - c In the navigation area, right-click `Configuration`, select **New > DWORD (32 bit) Value**, and type `RunScriptsOnStartSession`.
 - d In the topic area, right-click the entry for the new DWORD value and select **Modify**.

- e In the **Value data** text box, type 1 to enable start session scripting and click **OK**.
You can type 0 to disable this feature. The default value is 0.
 - f (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.
- 6 Set a registry value to specify timeout values in seconds rather than minutes to prevent scripts from timing out.
- Setting this timeout value in seconds enables you to configure the VMware View Script Host service timeout value in seconds. For example, if you set the VMware View Script Host service timeout to 30 seconds, you can ensure that a start session script either finishes running or times out before a View Connection Server timeout occurs.
- a Navigate to HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents.
 - b Add a DWORD value called TimeoutsInMinutes.
 - c Set a data value of 0.
- 7 (Optional) To enable the VMware View Script Host service to time out the start session script, set a timeout value.
- a Navigate to HKLM\SOFTWARE\VMware, Inc.\VMware VDM\ScriptEvents\StartSession.
 - b In the topic area, right-click the Default (@) key and select **Modify**.
 - c In the **Value data** text box, type the timeout value and click **OK**.
A value of 0 means that no timeout is set.
- 8 Exit the Registry Editor and restart the system.

Sample Start Session Scripts

These sample start session scripts illustrate how to write environment variables to a file, test the timeout functionality, and test a non-zero exit code.

The following sample Visual Basic script writes all the environment variables provided to the script into a file. You can use this sample script to see example data in your own environment. You might save this script as C:\sample.vbs.

Option Explicit

Dim WshShell, FSO, outFile, strOutputFile, objUserEnv, strEnv

strOutputFile = "c:\setvars.txt"

Set FSO = CreateObject("Scripting.FileSystemObject")

Set outFile = FSO.CreateTextFile(strOutputFile, TRUE)

outFile.WriteLine("Script was called at (" & Now & ")")

Set WshShell = CreateObject("WScript.Shell")

Set objUserEnv = WshShell.Environment("PROCESS")

For Each strEnv In objUserEnv

```
outFile.WriteLine(strEnv)  
Next
```

```
outFile.Close
```

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
```


Index

A

- Active Directory users and groups **41**
- advanced PowerShell commands **45**
- alerts **73**
- assigning a name **65**
- automatically provisioned desktop pools **40**

C

- components **7**
- connection broker events **11**
- connectivity monitoring **73**

D

- database queries and views **25**
- database tables and schemas **9**
- datastore management **48**
- desktop entitlements **42**
- desktop policies **81**
- desktop pool management tasks **39**
- disabling network label assignments **58**
- discovery script **67**
- displaying discovered objects **72**

E

- error handling **29**
- escaping characters **29**
- event database **9**
- event message attributes **24**
- event reports **44**
- examining PCoIP statistics **80**
- exporting LDAP data **60**

G

- general PCoIP session statistics **76**
- global settings **44**
- glossary **5**

I

- importing LDAP configuration data **63**
- importing management packs **66**
- integration interfaces **8**
- intended audience **5**
- introduction **7**

L

- LDAP configuration data **59**
- LDAP data customization **59**
- LDIF configuration file entries **61**
- LDIF configuration file format **60**
- LDIFDE command **64**
- licenses **44**
- linked-clone desktop pool management **40**

M

- managed objects **68**
- management packs **66**
- manual unmanaged desktop pools **41**
- manually provisioned desktop pools **41**
- modifying LDAP configuration data **60**
- monitoring View **70**
- multiple network labels **54**

N

- network label assignments **56**
- network label information **57**
- network label configuration file **50, 51, 53**
- network labels **50, 55**
- NIC and network label information **52**

O

- object classes and relationships **68**

P

- PCoIP audio statistics **76**
- PCoIP imaging statistics **77**
- PCoIP network statistics **78**
- PCoIP session statistics **75**
- PCoIP USB statistics **79**
- performance information **72**
- physical machines **43**
- piping objects **29**
- proxy server **67**

R

- rebalance operation **55**
- recompose operation **55**
- remote session management **42**
- remote systems **28**

resizing pools **46**

S

SCOM integration **65**

start session scripts **81, 82, 84**

starting PowerShell **28**

V

vCenter Server management tasks **39**

vCenter Server network label assignments **57**

View Administrator events **17**

View Agent events **16**

View component services **73**

View Connection Server management **38**

View Connection Server service **45**

View PowerCLI cmdlets **27, 32, 34**

View PowerCLI comparison **30**

View PowerCLI cmdlet examples **38**

View PowerCLI help **28**

views and monitors for View **70**

virtual machine management **43**

virtual machine ownership **43**

VMware View Script Host service **82**

vSphere datastore objects **48**

vSphere inventory objects **47**

W

Windows registry entries **83**