# VDI 3.1 Administration Guide

Contents

# VDI 3.1 Administration Guide

Contents

# Desktop Provider Management (All Topics)

## How to Create Desktop Providers (Sun VirtualBox)

Desktop providers encapsulate the details of the underlying virtualization platform. At a minimum, you must configure one desktop provider before you can continue with the creation of pools. There is no limitation to the number of providers the system can manage. At any time, you can configure additional providers.

Follow the steps below to set up a VirtualBox desktop provider.

VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Desktop Providers category in the left sidebar.

3. Select New in the Sun VirtualBox Desktop Providers overview.
   The New Desktop Provider for Sun VirtualBox wizard is displayed. It enables you to add multiple VirtualBox hosts as well as multiple VirtualBox storage servers.
   a. Type the host name or IP address and the administrator credentials for the VirtualBox host server.
   b. When you are finished adding new VirtualBox hosts, select the Select Existing Hosts option.
   c. Type the host name or IP address and the administrator credentials for the VirtualBox storage server.
   For a demo setup, the storage host will be the same as the VirtualBox host. You can configure multiple VirtualBox storage servers.
   d. When you are finished adding new storage hosts, select the Select Existing Storage option.
   e. Click Finish.
   The new desktop provider is displayed in the VDI Manager. You can now view the provider details, including CPU and memory utilization. You can add or remove additional VirtualBox hosts or storage servers as needed.

# How to Create Desktop Providers (VMware vCenter)

Desktop providers encapsulate the details of the underlying virtualization platform. At a minimum, you must configure one desktop provider before you can continue with the creation of pools. There is no limitation to the number of providers the system can manage. At any time you can configure additional providers.

Follow the steps below to set up a VMware vCenter desktop provider.

Before You Begin

- Make sure that the two-year SSL certificate for the vCenter server has not expired by checking the details of the certificate in a browser that has accepted it previously.

VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Desktop Providers category in the left sidebar.

3. Select New in the VMware vCenter Desktop Providers overview.
   This will activate the New Desktop Provider for VMware vCenter wizard.
   a. Type the name or IP address and the administrator credentials for the VMware vCenter.
   b. Click Finish.
   The new desktop provider is displayed in the VDI Manager. You can now view the VMware vCenter resource details, including datacenters, VMware clusters, and datastores.

> In Sun VDI 2.0, you could limit the used VMware vCenter resources to specific VMware datacenters. This resource restriction is now handled as part of the pool configuration.

# How to Create Desktop Providers (Microsoft Hyper-V)

Desktop providers encapsulate the details of the underlying virtualization platform. At a minimum, you must configure one desktop provider before you can continue with the creation of pools. There is no limitation to the number of providers the system can manage. At any time, you can configure additional providers.

### Before You Begin

The Windows Server hosting Hyper-V must be prepared to communicate with the VDI Core before a desktop provider can be created. Refer to the How to Prepare a Windows Server for VDI page for detailed information.

### VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Desktop Providers category in the left sidebar.

3. Select New in the Microsoft Hyper-V Desktop Providers overview.
   The New Desktop Provider for Microsoft Hyper-V wizard is displayed. It enables you to add multiple Hyper-V hosts as well as multiple Hyper-V storage servers.
   a. Type the host name or IP address and the administrator credentials for the Hyper-V host server.
   b. When you are finished adding new Hyper-V hosts, select the Select Existing Hosts option.
   c. Type the host name or IP address and the administrator credentials for the Hyper-V storage server.
      You can configure multiple Hyper-V storage servers.
   d. When you are finished adding new storage hosts, select the Select Existing Storage option.
   e. Click Finish.
      The new desktop provider is displayed in the VDI Manager. You can now view the provider details, including CPU and memory utilization. You can add or remove additional MS Hyper-V hosts or storage servers as needed.


# How to Create Desktop Providers (Microsoft Remote Desktop Services)

Desktop providers encapsulate the details of the underlying virtualization platform. At a minimum, you must configure one desktop provider before you can continue with the creation of pools. There is no limitation to the number of providers the system can manage, but note that there can be only one pool per desktop provider. At any time, you can configure additional providers.

### Before You Begin

The Windows Server hosting Hyper-V must be prepared to communicate with the VDI Core before a desktop provider can be created. Refer to the How to Prepare a Windows Server for VDI page for detailed information.

### VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Desktop Providers category in the left sidebar.

3. Select New in the Microsoft Remote Desktop Providers overview.
   The New Desktop Provider for Microsoft Remote Desktop wizard is displayed. It enables you to add either a Microsoft Remote Desktop (Terminal) Services host or several Microsoft Remote Desktop (Terminal) Services hosts that all participate in the same cluster.
   a. Type the host name or IP address and the administrator credentials for the host.
   b. When you are finished adding hosts, add more hosts or select the Select Existing Hosts option.
   c. Click Finish.
      The new desktop provider is displayed in the VDI Manager. You can now view the provider details, including CPU and memory utilization. You can add or remove additional Microsoft Remote Desktop Services hosts as needed, provided they all belong to the same cluster.

# How to Create Automated Administration Scripts

The `/opt/SUNWvda/sbin/vda` CLI can be used in scripts for automated administration.

## Reading the Return Code

The `/opt/SUNWvda/sbin/vda` returns the following exit codes:

- 0: Successful completion
- 1: An error occurred
- 2: Invalid command line options or arguments were specified

## Waiting for a Job to Finish

Some `vda` subcommands return immediately but start an action in the background, a job.
The subcommand `job-wait` allows to synchronously wait for a specific job to be completed.

```
# /opt/SUNWvda/sbin/vda job-wait --help
Wait until the job ends

Usage:
 vda job-wait [-t <timeout> | --timeout=<timeout>] <job>
 -?, --help               Print this help list
Options:
 -t <timeout>, --timeout=<timeout>
                          Timeout in seconds to wait
Operand:
*<job>                    The id of the job
'*' denotes mandatory parameters.
```

## Parsing the Output of the CLI

A number of subcommands support a parseable option so that the output is formatted for easy parsing: as a list of lines of colon-separated (':') fields.

The syntax of the option is:

```
 -x, --parseable        Display output suitable for programmatic parsing.
```

### user-search

Search for users/groups in the user directory that match the specified search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Name of the user/group | string |
| Kind of object | `User` / `Group` |
| DN of the user/group | string |

### user-show

Show the desktops available for the user.

Parseable Output in the case of a user: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | `User` / `Token <token>` / `Group <group_name>` / `Custom Group <group_name>` |

Parseable Output in the case of a group: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |

### user-desktops

Show the desktops assigned to the user.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | `flexible` / `personal` |
| Is Default Desktop | `true` / `false` |

### group-list

Lists all custom groups.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Custom Group Name | string |

### group-show

Show the pools assigned to the custom group.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |

### token-search

Search for tokens that match the search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Token | string |
| Name of the Associated User | string |

| DN of the Associated User | string |
|---|---|

### token-show

Show the desktops available for the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | `User` / `Token` / `Group` `<group_name>` / `Custom Group` `<group_name>` |

### token-desktops

Show the desktops assigned to the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | `flexible` / `personal` |
| Is Default Desktop | `true` / `false` |

### pool-list

List all pools.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Type of Desktop Assignment | `Personal` / `Flexible` |
| Number of Desktops | integer |
| Desktop Provider Name | string |

### pool-show

Show detailed information about the pool.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Assignment Status | `Enabled` / `Disabled` |
| Type of Desktop Assignment | `Personal` / `Flexible` |
| Desktop Provider Name | string |

| | |
|---|---|
| Cloning Status | `Enabled` / `Disabled` |
| Template | `None` / string |
| Number of Cloning Jobs | integer |
| Number of Available Desktops | integer |
| Number of Assigned Desktops | integer |
| Total Number of Desktops | integer |

## pool-desktops

List all desktops from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | long |
| Machine State | `Running` / `Powered Off` / `Suspended` / `Unknown` |
| Desktop State | `Available` / `Used` / `Idle` / `Unresponsive` / `Reserved` / etc. |
| DN of Assigned User | string |

## pool-templates

List all templates from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | long |
| Machine State | `Running` / `Powered Off` / `Suspended` / `Aborted` / `Unknown` |
| Master Revision | string |
| Cloned Desktops | string |

## template-revisions

List the revisions of the template.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Revision Name | string |
| Revision ID | long |
| Creation Date | timestamp |
| Is It Master | `yes` / `no` |
| Cloned Desktops | string |

## provider-list

List all desktop providers.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Provider Name | string |
| Provider Type | `Sun VirtualBox` / `VMware vCenter` / `Microsoft Hyper-V` / `Microsoft Remote Desktop` |
| Total Number of Desktops | integer |
| Number of Used Desktops | integer |
| CPU Usage | `xx% (x.x GHz/MHz)` |
| Memory Usage | `xx% (x.x GB/MB)` |
| Storage Usage | `xx% (x.x GB/MB)` |

### provider-list-hosts

List all hosts for the VirtualBox desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Host Name | string |
| Status | `OK` / `Unresponsive` / etc. |
| Enabled | `Enabled` / `Disabled` |
| CPU Usage | `xx% (x.x GHz/MHz)` |
| Memory Usage | `xx% (x.x GB/MB)` |
| Number of Desktops | integer |

### provider-list-storage

List all storage servers for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Status | `OK` / `Unresponsive` / etc. |
| Enabled | `Enabled` / `Disabled` |
| ZFS Pool | string |
| Capacity | `xxx.x GB` |
| Usage | `xx.x GB` |
| Number of Desktops | integer |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Storage ID | string |

| | |
|---|---|
| ZFS Pool | string |
| Capacity | `xxx.x GB` |
| Usage | `xx.x GB` |
| Number of Desktops | integer |

### provider-list-templates

List the templates for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | string |
| Path | string |

### provider-list-unmanaged

List the desktops from the virtualization platform that are not managed by any desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Host Name | string |
| Desktop Name | string |
| Desktop ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | string |

### provider-list-networks

List all networks for the desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Subnet Label | String |
| Subnet Address | String |
| Availability | `All Hosts` / `Not on: <comma_separated_list_of_hosts>` |

## job-list

List the existing jobs.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| ID of the Job | integer |
| Cancellable | 'C' if the job can be cancelled |

## job-show

Show the job details.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| Start Time | hh:mm:ss |
| End Time | hh:mm:ss |
| Job Details | string |
| Cancellable | `true` / `false` |

## Contents

# Desktop Pool Management (All Topics)

## About New Pool Settings

In VDI, a pool is defined as a collection of desktops. With VDI 3.1, you can take advantage of configuration settings that are only applied to a selected pool, managed by the VDI Core.

### Per Pool RDP Configuration

With VDI 3.1, you can configure the RDP options to be used by Sun Ray sessions when users connect to their desktops. A wide range of options including locale, keyboard layout, color depth, theming, redirection and more are configurable on a per pool basis using the VDI administration console or CLI. These settings will be used by Sun Ray sessions connecting to desktops managed by any of the currently supported VDI desktop provider types.

For details on how to enable this feature, see the How to Configure RDP Options Per Pool page.

### About the Smart Card Removal Policy

With VDI 3.1, you can control what should happen to a user's desktop after a smartcard is removed from a Sun Ray DTU. Using the Smart Card Removal Policy, you can indicate that a user's desktop should be shut down, suspended, or recycled when the smart card has been out of a DTU for a specific length of time. If the user reinserts a smart card before the specified time has elapsed, the associated action on the desktop will be canceled. The Smart Card Removal Policy is configurable per pool and is available for all Sun VirtualBox, Microsoft Hyper-V, and VMware vCenter pools. This policy may be configured using the VDI Manager or CLI.

> ℹ️ Recycling is applied only to desktops that have flexible assignments. Choosing the recycle option for your Smart Card Removal Policy will have no effect on personally assigned desktops.

For details on how to enable this feature, see the How to Configure Smart Card Removal page.

### About Running Available Desktops

With VDI 3.1, you can take advantage of faster access to desktops by using the Running Available Desktops feature. For each of your desktop pools, you can ensure that any number of available desktops will be maintained in the running state. This can considerably reduce the time taken to deliver desktop sessions to your users.

## About Per Pool Network Configuration

In VDI 3.1 you can take advantage of the Per Pool Network Configuration feature, which enables an administrator to specify the subnet in which desktops will be placed. For Sun VirtualBox and Microsoft Hyper-V desktop providers, VDI will detect the networks that are configured on the provider's hosts, and the administrator can select which of these networks should be used in specific pools.

Configuration of networks is done at two levels in VDI:

- Desktop Provider - Each subnet available on either a Sun VirtualBox or Microsoft Hyper-V host is identified by a unique label. By default this label is the subnet address, but it can be changed in the Network tab for the desktop provider. When a host is added to a desktop provider, VDI will detect the subnets available on that host and will update the Network table accordingly. If a subnet is not available on any of the hosts in a provider, VDI will display a warning. You can view the list of subnets available for a specific host by selecting that host in the Host tab for the desktop provider. If you make changes to the networking on a host, click the Refresh button in the Network tab so that VDI can rescan the subnets available on the host.

- Desktop pool - A pool can have one or more networks assigned to it. When a pool is created, VDI will check whether any networks are available on all hosts for the desktop provider of the pool, and it will assign one of these networks to the pool. If no networks are available on all hosts for the provider, the administrator must explicitly specify a network to be used by the pool through the Settings tab for the pool. When desktops are imported or cloned in a pool, VDI will create a network device on the desktop and configure that device to be in the networks that have been enabled for the pool. If more than one network has been configured for the pool, VDI will

use the network that has been configured as the primary network when trying to establish an RDP connection to the desktop. The primary network for a pool can be configured in the Settings tab.

> ℹ The Per Pool Network Configuration feature is only available for Sun VirtualBox pools if Host Networking is being used.

# How to Create Desktop Pools

Sun VDI organizes desktops in pools. A pool is a collection (or container) of desktops. Typically you will create different pools for different types of users. For example, the engineering team at your company might have different desktop requirements than the marketing department.

> ⛔ Sun VirtualBox Desktop Providers Only
> When changing pool settings from NAT networking to Host Networking + Windows RDP, existing desktops that are running must be stopped and restarted or else subsequent user requests for these desktops will fail.
> This issue occurs because existing, running desktops will be using NAT and will not have a public IP address. After the pools settings have been changed, subsequent requests for that desktop will attempt to access the desktop via the private (and unaccessible) NAT IP.

> ℹ Microsoft Remote Desktop Providers Only
> Only one pool can be created per Microsoft Remote Desktop provider.

VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Pools category in the left sidebar.

3. Click New in the All Pools overview.
   A New Pool wizard is displayed.

   a. For Sun VirtualBox and Microsoft Hyper-V desktop providers, choose one of the following pool types:
      - Dynamic pools are filled with cloned flexible desktops. If you choose the Dynamic Pool type, the desktops in the pool will be temporarily assigned to users. They will be recycled each time the user logs out. This pool type is considered dynamic because the user-desktop assignments are often changing.
      - Growing pools are filled with cloned personal desktops. If you choose the Growing Pool type, the desktops in the pool will be permanently assigned to users. Users can log in and out without losing their desktop settings. The desktops are not recycled.
      - Manual pools are initially empty. They are filled manually by importing personal desktops. The Manual Pool type should be used if cloned desktop assignment is not an option.

      > ℹ For Microsoft Remote Desktop providers, pool types do not apply.

   b. Select a template.
      If you have already imported a desktop from Sun VirtualBox or Microsoft Hyper-V, you can select it as a template to clone desktops from.
      If no desktop has been imported yet, select None from the drop down menu. After a desktop has been imported, you can select it as a template from the pool's Cloning tab.
   c. If you chose a template in the previous step, select the pool size or enable automatic cloning.
      You can modify your choice at any time in the pool's Cloning tab.
   d. Click Finish.
      A new pool is displayed in the Pools overview.

# How to Configure Networks Per Pool

For Sun VirtualBox and Microsoft Hyper-V pools, you can select which network desktops imported or created in the pool will use. For more detailed information, see About Per Pool Network Configuration.

## VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Pools category, and click an existing pool.

3. Click the Settings tab to specify which networks are configured on the desktops in the pool.
   For each network selected, a network adapter will be created on new desktops created in that pool.

> - Rename or refresh the desktop provider network list - Select the Desktop Providers category, and select the Sun VirtualBox or Microsoft Hyper-V desktop provider of interest. Select the Network tab to see the network(s) configured on the desktop provider. After making changes to the networking on a Sun VirtualBox or Microsoft Hyper-V host, click the Refresh button to rescan the network list for the provider.
> - View a read-only list of the networks on a specific host - Go to the Desktop Providers category, and select a desktop provider. Then select the host in the Host tab.

# How to Configure RDP Options Per Pool

With VDI 3.1 you can configure the RDP options to be used by Sun Ray sessions when users connect to their desktops.

## VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Pools category and select the pool of interest.
   a. In the pool overview, select the Settings tab.
   b. In the Sun Ray section, click the Edit Sun Ray RDP Settings link.
   c. Enable the desired RDP settings and click Save.
   d. Click Back, and select the Use Customized Settings option in the Sun Ray section.
   e. Click Save.

## Available RDP Options

Sun Ray Connector for Windows OS (`uttsc`) supports a wide range of options allowing you to configure RDP connections from Sun Ray to your users' desktops.

VDI 3.1 allows you configure a subset of these options on a per pool basis. The following table lists the supported options. For details about how VDI's Sun Ray settings compare to the SRWC `uttsc` settings, refer to the VDI Defaults page.

| Name | Description | Default Value |
|------|-------------|---------------|
| General | | |

| | | |
|---|---|---|
| Locale | Use this setting to identify the locale used for users' desktop sessions.<br>Any valid locale identifer may be specified, for example, `en-US` or `de-DE`. | en-US |
| Keyboard Layout | Use this setting to identify the keyboard type used for users' desktop sessions<br>Valid values for this setting include All Sun and PC USB Keyboards, Sun Type6 Japanese Keyboard, and Sun Korean Keyboard. | All Sun and PC USB Keyboards |
| Optimized Hotdesking | Use this setting to enable or disable optimized hotdesking behaviour. If enabled, Sun Ray sessions can be hotdesked without restarting `uttsc`. | Disabled |
| Windows Pulldown Header | Use this setting to enable or disable the Windows pulldown header. | Enabled |
| RDP Packet Data Compression | Use this setting to enable or disable the compression of RDP packet data. | Enabled |
| Appearance | | |
| Colour Depth | Use this setting to specify the preferred colour depth for users' desktop sessions.<br>Valid values for this setting are 8, 15, 16, 24 and 32.<br>Note: Colour depth may be limited by configuration of the desktop to which a user connects. In such cases the available colour depths of the desktop will take priority over the colour depth configured for the pool containing the desktop. | 32 |
| Theming | Use this setting to enable or disable theming for users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Desktop Background | Use this setting to enable or disable the desktop background for users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Show Window Contents While Dragging | Use this setting to enable or disable the ability to show complete window contents while dragging windows in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Transition Effects for Menus | Use this setting to enable or disable visual effects during the use of menus in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Pointer Shadow | Use this setting to enable or disable the use of pointer shadow in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Pointer Scheme | Use this setting to enable or disable the use of pointer schemes in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Sound | Use this setting to control sound quality in users' desktop sessions.<br>Valid values for this setting are "High" (to enable high quality sound), "Low" (to enable low quality sound) and "Off" (to disable sound). | High |
| Redirection | | |
| Smart Cards | Use this setting to enable or disable smart card redirection from a DTU to users' desktop sessions. | Disabled |
| USB | Use this setting to enable or disable USB redirection from a DTU to users' desktop sessions. | Enabled |
| Serial Devices | Use this setting to identify serial devices which should be redirected to users' desktop sessions.<br>Valid values for this setting are specified using the format <comport>=<device> where <device> identifies the serial device to be redirected and <comport> identifies the port (on the users' desktops) that <device> should be redirected to. | No serial devices are redirected by default. |
| Paths | Use this setting to identify paths (available on a VDI host) which should be redirected to drives on users' desktop sessions.<br>Valid values for this setting are specified using the format <drive name>=<path> where <path> identifies the path to be redirected and <drive name> identifies the drive (on the users' desktops) that <path> should be redirected to. | No paths are redirected by default. |

| Printers | Use this setting to identify printer queues which should be redirected to users' desktop sessions.<br><br>Valid values for this setting are specified using the format <printer>=[<driver>] where <printer> identifies the printer queue to be redirected and <driver> identifies a printer driver to be used for the printer on users' desktop sessions. If <driver> is omitted, a simple PostScript driver is used by default. | No printer queues are redirected by default. |

# How to Configure Smart Card Removal

With VDI 3.1 you can control what should happen to a user's desktop after a smart card is removed from a Sun Ray DTU. For more information, see About New Pool Settings.

## VDI Manager Steps

1. Sign into the VDI Manager.
    a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
    b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Navigate to the Pool Settings tab for the pool to be configured.
    a. In the Sun Ray section, indicate the action you want to be associated with removal of smart cards from DTUs using the Action on Card Removal menu.
        • No Action - Select if you want VDI to ignore smart card removals.
        • Recycle Desktop - Select if you want flexibly assigned desktops to be recycled.
        • Shutdown Desktop - Select if you want desktops to be shut down.
        • Suspend - Select if you want desktops to be suspended.
    b. Specify the number of seconds a smart card must be removed from a DTU before any action should be taken in the Delay Action field.
    c. Click Save.

## CLI Steps

1. Open a terminal window and sign into the server with root credentials.
   For a multi-host configuration, use one of the VDI Secondary hosts.

2. Configure the desktop action associated with smart card removal.

```
# /opt/SUNWvda/sbin/vda pool-setprops -p card-removed=<desktop action> <pool name>
```

3. Specify the length of time (in seconds) that a smart card must be out of a DTU before the action is performed.

```
# /opt/SUNWvda/sbin/vda pool-setprops -p card-removed-timeout=<time in seconds> <pool name>
```

• Example – Specify the desktop action that should be performed after a smart card has been out of a DTU for a given amount of time

```
# /opt/SUNWvda/sbin/vda pool-setprops -p card-removed=suspend,card-removed-timeout=30 MyPool
```

# How to Enable USB Redirection

## Before You Begin

1. Prepare your Windows XP virtual machine template by installing the USB Redirector.
   See the How to Install the Sun Ray Connector Windows Components on the SRWC 2.2 information site for more details.

2. Add addtional USB drivers for virtual machines created in VMware vCenter or Microsoft Hyper-V.
   This step is not necessary for VirtualBox virtual machines. See How to Add USB Drivers to a VMware ESX or Hyper-V Server Virtual Machine on the SRWC 2.2 information site for more details.

Steps

1. Import the prepared virtual machine as a template into the VDI host.
   Refer to the following pages:
   - How to Import Desktops (Sun VirtualBox)
   - How to Import Desktops (Microsoft Hyper-V)
   - How to Import Desktops (VMware vCenter)

2. In Pool settings, select Edit RDP Settings. Save settings with USB enabled, select Use Customized RDP Settings, and save again.

3. (Optional) Clone some virtual machines with Sysprep enabled.

4. Once the virtual machine is available, obtain a session for any user, and log into the virtual machine.

5. Choose Computer -> Properties -> Hardware -> Device manager to see whether the driver is visible under USB Serial Bus Controllers.

The virtual machine is now ready to redirect any USB disk.

# How to Create Automated Administration Scripts

The `/opt/SUNWvda/sbin/vda` CLI can be used in scripts for automated administration.

## Reading the Return Code

The `/opt/SUNWvda/sbin/vda` returns the following exit codes:

- 0: Successful completion
- 1: An error occurred
- 2: Invalid command line options or arguments were specified

## Waiting for a Job to Finish

Some `vda` subcommands return immediately but start an action in the background, a job.
The subcommand `job-wait` allows to synchronously wait for a specific job to be completed.

```
# /opt/SUNWvda/sbin/vda job-wait --help
Wait until the job ends

Usage:
 vda job-wait [-t <timeout> | --timeout=<timeout>] <job>
 -?, --help               Print this help list
Options:
 -t <timeout>, --timeout=<timeout>
                          Timeout in seconds to wait
Operand:
*<job>                    The id of the job
'*' denotes mandatory parameters.
```

## Parsing the Output of the CLI

A number of subcommands support a parseable option so that the output is formatted for easy parsing: as a list of lines of colon-separated (':') fields.

The syntax of the option is:

```
    -x, --parseable          Display output suitable for programmatic parsing.
```

## user-search

Search for users/groups in the user directory that match the specified search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Name of the user/group | string |
| Kind of object | `User` / `Group` |
| DN of the user/group | string |

## user-show

Show the desktops available for the user.

Parseable Output in the case of a user: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | `User` / `Token <token>` / `Group <group_name>` / `Custom Group <group_name>` |

Parseable Output in the case of a group: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |

## user-desktops

Show the desktops assigned to the user.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | `flexible` / `personal` |
| Is Default Desktop | `true` / `false` |

## group-list

Lists all custom groups.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|

| | |
|---|---|
| Custom Group Name | string |

### group-show

Show the pools assigned to the custom group.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |

### token-search

Search for tokens that match the search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Token | string |
| Name of the Associated User | string |
| DN of the Associated User | string |

### token-show

Show the desktops available for the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | `User` / `Token` / `Group` `<group_name>` / `Custom Group` `<group_name>` |

### token-desktops

Show the desktops assigned to the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | `flexible` / `personal` |
| Is Default Desktop | `true` / `false` |

### pool-list

List all pools.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Type of Desktop Assignment | Personal / Flexible |
| Number of Desktops | integer |
| Desktop Provider Name | string |

## pool-show

Show detailed information about the pool.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Assignment Status | Enabled / Disabled |
| Type of Desktop Assignment | Personal / Flexible |
| Desktop Provider Name | string |
| Cloning Status | Enabled / Disabled |
| Template | None / string |
| Number of Cloning Jobs | integer |
| Number of Available Desktops | integer |
| Number of Assigned Desktops | integer |
| Total Number of Desktops | integer |

## pool-desktops

List all desktops from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | long |
| Machine State | Running / Powered Off / Suspended / Unknown |
| Desktop State | Available / Used / Idle / Unresponsive / Reserved / etc. |
| DN of Assigned User | string |

## pool-templates

List all templates from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | long |
| Machine State | Running / Powered Off / Suspended / Aborted / Unknown |

| | |
|---|---|
| Master Revision | string |
| Cloned Desktops | string |

### template-revisions

List the revisions of the template.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Revision Name | string |
| Revision ID | long |
| Creation Date | timestamp |
| Is It Master | `yes` / `no` |
| Cloned Desktops | string |

### provider-list

List all desktop providers.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Provider Name | string |
| Provider Type | `Sun VirtualBox` / `VMware vCenter` / `Microsoft Hyper-V` / `Microsoft Remote Desktop` |
| Total Number of Desktops | integer |
| Number of Used Desktops | integer |
| CPU Usage | `xx% (x.x GHz/MHz)` |
| Memory Usage | `xx% (x.x GB/MB)` |
| Storage Usage | `xx% (x.x GB/MB)` |

### provider-list-hosts

List all hosts for the VirtualBox desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Host Name | string |
| Status | `OK` / `Unresponsive` / etc. |
| Enabled | `Enabled` / `Disabled` |
| CPU Usage | `xx% (x.x GHz/MHz)` |
| Memory Usage | `xx% (x.x GB/MB)` |
| Number of Desktops | integer |

### provider-list-storage

List all storage servers for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Status | OK / Unresponsive / etc. |
| Enabled | Enabled / Disabled |
| ZFS Pool | string |
| Capacity | xxx.x GB |
| Usage | xx.x GB |
| Number of Desktops | integer |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Storage ID | string |
| ZFS Pool | string |
| Capacity | xxx.x GB |
| Usage | xx.x GB |
| Number of Desktops | integer |

## provider-list-templates

List the templates for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | string |
| Path | string |

## provider-list-unmanaged

List the desktops from the virtualization platform that are not managed by any desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Host Name | string |
| Desktop Name | string |

| | |
|---|---|
| Desktop ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | string |

### provider-list-networks

List all networks for the desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Subnet Label | String |
| Subnet Address | String |
| Availability | `All Hosts` / `Not on: <comma_separated_list_of_hosts>` |

### job-list

List the existing jobs.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| ID of the Job | integer |
| Cancellable | 'C' if the job can be cancelled |

### job-show

Show the job details.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| Start Time | hh:mm:ss |

| End Time | hh:mm:ss |
|---|---|
| Job Details | string |
| Cancellable | `true` / `false` |

Contents

# Desktop Management (All Topics)

## About Desktop and Virtual Machine States

In Sun Virtual Desktop Infrastructure (VDI), a user is assigned to one or several virtual desktops and can use these desktops from everywhere as if they were running on a traditional personal computer. Sun VDI provides advanced management and lifecycle features which allow the effective management of thousands of desktops. Desktops transition through states defined by settings in the VDI Core.

Virtual machines are used to run the operating systems which render the desktops. They are controlled by a hypervisor, such as Sun VirtualBox, Microsoft Hyper-V, and VMware Infrastructure. They cycle through traditional machine states such as powered off and running.

## Virtual Machine States

Virtual machine states are defined by the hypervisor. For Sun VDI 3.1, you have the choice of using a Sun VirtualBox, VMware Infrastructure, or Microsoft Hyper-V virtualization platform.

Sun VirtualBox and Microsoft Hyper-V

- Powered Off
  Powered off virtual machines reside in two places in the VDI environment, the database and the storage. The VDI database contains the desktop configuration information to register the desktop on a hypervisor. The storage server contains the desktop's hard disk data. Powered off virtual machines are typically not associated or registered on any hypervisor host. This strategy enables the VDI Core to select the best suited host on every start of a virtual machine. This setup helps ensure a distribution of virtual machines across available VirtualBox or Hyper-V hosts minimizing resource usage on each.

- Running
  Running desktops are registered and started on a single hypervisor host. The host that a virtual machine is running on can be determined using the Desktop Summary page in the VDI Manager. A running virtual machine is connected directly to the storage.

- Suspended
  Suspended virtual machines have been suspended by the hypervisor.

- Stuck, Aborted, Paused
  These machine states are specific to VirtualBox.

- Unknown
  This state typically indicates that the vCenter server cannot be contacted to retrieve the state information.

## Desktop States

The desktop states are used to accomplish the following:

- Implement the desktop lifecycle.
- Synchronize VDI hosts and desktop hosts.
- Serve as a tool for monitoring and analyzing the system state.

The following figure depicts a simplified version of the lifecycle of a flexibly assigned desktop.
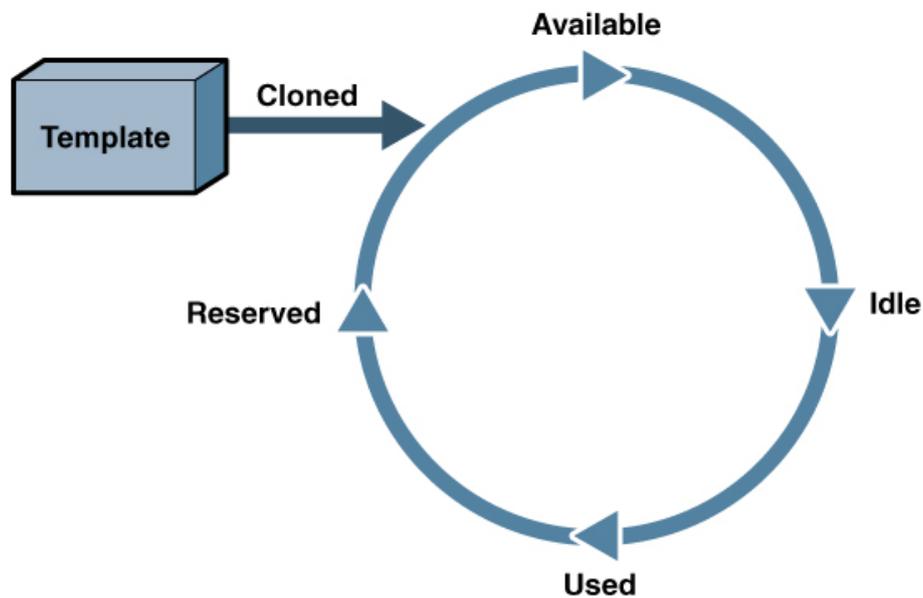


Possible desktop states are:

- Available - The first state
  A desktop is added to the database and then set to the Available state after being cloned from a template. After becoming Available, the desktop is ready to be assigned to users. If the recycle policy is set to Reuse Desktop or Reset to Snapshot, the desktop will return to this state.

- Idle - The intermediate state
  The desktop is in this state whenever the desktop is assigned and the user is not using it, for example, when the desktop is assigned and the user has not logged in yet or when the desktop is assigned and the user just logged out. A desktop is recycled after it remains in that state for a configurable amount of time.

  The VMware vCenter desktop provider has two additional Idle states: when the desktop is assigned and either the virtual machine is suspended or the guest OS goes into standby through the vCenter option Keep VM Running on Guest OS Standby.

- Used - The active state
  A desktop enters the Used state as soon as the user has logged in to the desktop. The desktop stays in this state while the user logs in, uses the desktop, and logs out.

- Reserved - The maintenance state
  A desktop is Reserved when it is being worked on by the VDI Core. This desktop state usually occurs when the desktop is the source of a manual copy operation or the desktop is recycled. The desktop will become Available after leaving the Reserved state.

- Unresponsive - The quarantine state
  The desktop enters the Unresponsive state whenever the VDI core determines a severe problem with the desktop. An unresponsive

desktop is outside the desktop life cycle and needs the attention of the administrator. The administrator may either fix the problem and apply the Activate action to the desktop, which puts the desktop back in the lifecycle, or the administrator may choose to delete the desktop.

# How to Create Virtual Machines (Sun VirtualBox)

Sun VDI presents users with easy access to their virtual desktops, instances of any desktop operating system executed in a virtual machine. You can manually create virtual machines, or you can configure Sun VDI to create or clone additional virtual machines automatically from a template.

## Before You Begin

After executing the Sun VirtualBox for VDI 3.1 installer, you can create your first virtual machine. To simplify the installation, use the installation of Sun VirtualBox for VDI 3.1 (described in the previous section) to create the first virtual machine template. You can also install a version of Sun VirtualBox for VDI locally (on your laptop or desktop system), and create virtual machines there. Make sure to install the same version of VirtualBox locally that you have installed on your VirtualBox server. You can download the supported local version of VDI from the VirtualBox for VDI 3.1 download site. See the VDI 3.1 Release Notes for addition version support information.

## Steps

1. Launch the Sun VirtualBox Web Console.

```
# /opt/VirtualBox/VirtualBox
```

   a. Click New to launch the New Virtual Machine wizard.
   b. The wizard will guide you through virtual machine creation.
      Be sure to choose the appropriate hard-disk and RAM space for the desired configuration (4 GB hard-disk and 384 MB RAM are recommended).
      For more information about virtual machine system requirements, refer to Chapter 3: Starting out with VirtualBox in the Sun VirtualBox User Manual.

2. Install the operating system.
   At this point you have an empty virtual machine, equivalent to a PC without an OS installed. The next step is to choose the boot medium for the OS and install it.
   a. Select the newly created virtual machine and click Settings.
   b. Open the Advanced tab in the Settings GUI.
   c. Ensure that CD/DVD-ROM is set as the first boot device.
   d. Select the CD/DVD-ROM option in the left panel of the Settings dialog.
   e. Select the Mount CD/DVD Drive option.
   f. Click OK to save the changes and close the Settings GUI.
      At this point the new virtual machine must be started to trigger the OS installation.
   g. Select the new virtual machine and click Start.
   h. Follow the installation prompts, or seek further installation details from the OS manufacturer.

3. Install the VirtualBox Guest Additions.
   After the OS is installed, the VirtualBox Guest Additions must be installed.
   a. With the virtual machine running and fully booted, click the Devices menu, then select Install Guest Additions.
      This will launch the VirtualBox Guest Additions installer inside the virtual machine.
   b. Install the Guest Additions according to the instructions in the wizard and reboot when asked.
   c. Install all additional software for the desired virtual machine template.

4. Enable System Preparation for Windows templates.
   For more information, refer to the How to Enable System Preparation for Windows Templates (VirtualBox and Hyper-V) page.

# How to Create Virtual Machines (VMware vCenter)

Sun VDI presents users with easy access to their virtual desktops, typically instances of Microsoft Windows XP executed in a virtual machine. You can manually create virtual machines, or you can configure Sun VDI to create or clone additional virtual machines automatically from a

template.

Steps

1. Create a virtual machine with Microsoft Windows.
   Use your standard process for creating virtual machines. For full details, see "Creating Virtual Machines" in VMware Basic System Administration.
   Follow these recommendations:
   - Use Microsoft Windows XP SP3 as the baseline. The license must be a volume license.
   - Define one disk. It should be as small as possible, for example, 4 GB. The size impacts system performance and overall storage consumption.
   - RAM also should be as small as possible, for example, 384 MB.
   - A single CPU should be enough.
   - One network interface is needed. It should be configured for DHCP. Ensure that the virtual machine obtains a valid IP after powering on.

2. Install the VMware Tools.
   Once you have created a virtual machine with Microsoft Windows XP installed on it, install VMware Tools. VMware Tools is a suite of utilities that enhances the performance of the virtual machine's guest operating system and improves management of the virtual machine. Installing VMware Tools in the guest operating system is vital.
   The installation can be easily triggered from within the VMware Virtual Infrastructure Client (VIC): Right-click the virtual machine and choose Install VMware Tools. For additional details see Installing and Upgrading VMware Tools in VMware Basic System Administration.

3. Enable remote desktop access.
   RDP is the main access method to the Microsoft Windows XP desktop. By default, this access method is disabled and rejected through the firewall. To enable remote desktop access, launch VMware's Virtual Infrastructure Client, with your virtual machine still powered on and logged in, then follow these steps:
   a. Open a console for the virtual machine, and click the virtual machine's Start button.
   b. Right-click on My Computer in the start menu, and select Properties.
   c. In the System Properties window, select the Remote tab.
   d. Under Remote Desktop, check the box marked Enable Remote Desktop on This Computer so that this item is selected.
   e. Make sure that the desired users have been granted remote access rights.
   f. Click OK to save the settings and close the dialog.
      Before you try to connect to a virtual desktop remotely, ensure that no firewall blocks the remote access. Make sure that port 3389 is enabled in any firewall that may be active on the system.

4. Install the Sun VDA Tools.
   VDI has a tools component that notifies the VDI service when a desktop is in use and handles RDP connections when the guest OS initiates Standby. The VDI Tools must be installed on the guest operating system for recycling to work correctly and so that the RDP connection is correctly closed when the virtual machine goes into Standby or Suspend mode.
   a. Locate the `vda-tools.msi` installer file in the directory where you unzipped the VDI archive.
      The `vda-tools.msi` is located in the `./image/vda_3.1/Windows/Packages/` subdirectory. Copy the installer to the desired virtual machine.
   b. Within the virtual machine's console, double-click the installer and follow the prompts to complete installation.
      The default target location for the VDI Tools on Windows is `C:\Program Files\Sun\Virtual Desktop Access\Tools`.
   c. The VM Services list should now contain a new service named Sun VDI Tools, running and set to start automatically.

5. Configure power management.
   An unused virtual desktop is treated like a laptop that is unplugged to save power. It is suspended automatically to release all its CPU and memory consumption. Suspending virtual machines also affects how they are recycled. A virtual machine will be recycled if it has been suspended for longer than the recycling idle timeout period, even if the user has not logged out of the machine.
   a. Open the Virtual Infrastructure Client.
   b. Select the desired virtual machine and open the console.
   c. Log into the virtual machine.
   d. Go to Start -> Control Panel. Open Power Options.
   e. Set the System Standby time to the desired value.
   f. Verify that the guest OS actually enters standby as configured.
      Virtual machines should be configured to be suspended when the OS goes into standby. This is enabled in VMware vCenter.
   g. Open the Virtual Infrastructure Client.
   h. Right-click on the desired virtual machine and go to Edit Settings
   i. Go to Options -> Power Management, select Suspend the Virtual Machine.

6. Convert a virtual machine into a template.
   You can clone additional virtual machines manually, or let Sun VDI clone them automatically from a template. Any existing virtual

machine can be converted into a template.
- a. Open the Virtual Infrastructure Client.
- b. Right-click the desired virtual machine and power down the machine.
- c. From the commands area or the pop-up menu, click Convert to Template.
  For additional details see Chapter 13: Working with Templates and Clones in VMware Basic System Administration.

7. Create a Customization Specification.
   It is necessary to customize the identity and network settings of Windows XP after a clone has been created from a template. This can be achieved using a Customization Specification.
   - a. Open the Virtual Infrastructure Client.
   - b. Click Edit from the menu above the tool bar and select Customization Specifications.
   - c. Click the New icon in the Customization Specification Manager to start the wizard.
   - d. On the first wizard step, choose Windows as the target virtual machine OS, and give the specification a name and description.
   - e. The following steps ask the standard Windows installation questions and should be completed to correspond with your requirements, with the exception of the following.
     - Computer Name: Make sure that the Use the Virtual Machine Name item is selected. If not, you may end up with duplicate hostnames.
     - Windows License: Enter your Windows XP serial number. The Include Server License Information item should be left unchecked.
     - Networking: Make sure the interface is configured for DHCP. If not, your cloned virtual machines will not have unique IP addresses and will not work with Sun VDI.
   - f. After completing the wizard and saving your customization specification, close the Customization Specification Manager.
     For additional details see Chapter 14: Customizing Guest Operating System in VMware Basic System Administration.

# How to Create Virtual Machines (Microsoft Hyper-V)

Sun VDI presents users with easy access to their virtual desktops, typically instances of Microsoft Windows XP executed in a virtual machine. You can manually create virtual machines, or you can configure Sun VDI to create or clone additional virtual machines automatically from a template.

## Steps

1. Create a virtual machine with Microsoft Windows.
   Use your standard process for creating virtual machines. For information on how to create a virtual machine in Microsoft Hyper-V, refer to the Microsoft documentation.
   Follow these recommendations:
   - Use Microsoft Windows XP SP3 as the baseline. The license must be a volume license.
   - Define one disk. It should be as small as possible, for example, 4 GB. The size impacts system performance and overall storage consumption.
   - RAM also should be as small as possible, for example, 384 MB.

2. Install the Microsoft Hyper-V Integration Components.
   Once you have created a virtual machine with Microsoft Windows XP installed on it, install the Hyper-V Integration Components. The Integration Components allow Microsoft Hyper-V and Sun VDI to interoperate with the virtual machine. Installing the Integration Components in the guest operating system is vital.
   The installation can be easily triggered from within the Hyper-V Management Console: Connect to the virtual machine from the console and select the Insert Integration Services Setup Disk option from the Action menu.

3. Enable remote desktop access.
   RDP is the main access method to the Microsoft Windows XP desktop. By default, this access method is disabled and rejected through the firewall. To enable remote desktop access, connect to the virtual machine from the Hyper-V Management Console and follow these steps:
   - a. In the console, click the virtual machine's Start button.
   - b. Right-click on My Computer in the start menu, and select Properties.
   - c. In the System Properties window, select the Remote tab.
   - d. Under Remote Desktop, check the box marked Enable Remote Desktop on this computer so that this item is selected.
   - e. Make sure that the desired users have been granted remote access rights.
   - f. Click OK to save the settings and close the dialog.
     Before you try to connect to a virtual desktop remotely, ensure that no firewall blocks the remote access:
     Make sure that port 3389 is enabled in any firewall that may be active on the system.

4. Install the Sun VDA Tools.

Sun VDI 3.1 has a tools component that notifies the VDI service when a desktop is in use and handles RDP connections when the guest OS initiates Standby. The VDA Tools must be installed on the guest operating system for recycling to work correctly and so that the RDP connection is correctly closed when the virtual machine goes into Standby or Suspend mode.

    a. Locate the `vda-tools.msi` installer file in the directory where you unzipped the VDI archive.
       The `vda-tools.msi` is located in the `/var/tmp/vda_3.1_amd64/vda_3.1/Windows/Packages` subdirectory. Copy the installer to the desired VM.

    b. Within the VM's console double-click the installer and follow the prompts to complete installation.
       The default target location for the VDA Tools on Windows is `C:\Program Files\Sun\Virtual Desktop Access\Tools`.

    c. The VM services list should now contain a new service named Sun VDA Tools, running and set to start automatically.

# How to Import Desktops (Sun VirtualBox)

A pool is empty and has no desktops after initial creation. After you create virtual machines, you must import them into the VDI Core database.

> ⚠️   Importing snapshots of virtual machines is not supported.

### Before You Begin

A virtual machine must be created in the VirtualBox interface or using the integrated VDI Manager Flash console before it can be imported into the VDI Core database. Refer to the How to Create Virtual Machines (Sun VirtualBox) page for detailed information.

### VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Open the Pools tab, then select the previously created empty pool.

3. Select the Desktops tab, and click Import.
   An import dialog will be displayed.

4. Select a desktop to be imported.
   - If the desktop you would like to import is on the VirtualBox host, select it from the VirtualBox Host tab, and click OK.
   - If the desktop you would like to import is available on the VDI host in `/var/tmp`, define the corresponding XML and VDI files under the VDI Host tab.

After the desktop has been imported successfully, it will be displayed in the Desktops tab of the Pools page. A page refresh might be necessary.

> ℹ️   You can move your virtual machine XML file and VDI image to `/var/tmp` using the following command:
>
> ```
> # scp <path to file> root@<VDI host>:/var/tmp
> ```
>
> If there is not enough space in `/var/tmp`, you can copy both files to another location on the host and create a symbolic link:
>
> ```
> # ln -s <path to file> /var/tmp/
> ```

# How to Import Desktops (VMware vCenter)

A pool is empty and has no desktops after initial creation. After you create virtual machines, you must import them so that the VDI Core can

create a corresponding entry for the virtual machine in its database. The virtual machine will not be altered in any way.

Before You Begin

A virtual machine must be created in VMware vCenter before it can be imported into the VDI Core. Refer to the How to Create Virtual Machines (VMware vCenter) page for detailed information.

VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Open the Pools tab, then select the previously created empty pool.

3. Select the Desktops tab, and click Import.
   An import dialog is displayed showing the available virtual machines in the vCenter hierarchy. You can select individual virtual machines or folders. If you select a folder, all the virtual machines in the folder will be selected for the import.

4. Click OK to import the desktops into the VDI Core database.

> ℹ  Desktops that are already imported into the VDI Core cannot be selected for import. You also cannot import templates.

After the desktops have been imported successfully, they will show up in the Desktops tab of the Pool page (a page refresh might be necessary).


# How to Import Desktops (Microsoft Hyper-V)

A pool is empty and has no desktops after initial creation. After you create virtual machines, you must import them into the VDI Core database.

> ⚠  Importing snapshots of virtual machines is not supported.

Before You Begin

A virtual machine must be created in Microsoft Hyper-V before it can be imported into the VDI Core database. Refer to the How to Create Virtual Machines (Microsoft Hyper-V) page for detailed information.

VDI Manager Steps

1. Export the virtual machine template from the Hyper-V server.
   a. In the Hyper-V management console, select the Hyper-V virtual machine template.
   b. Select Export from the Actions menu and choose a directory on the Hyper-V server to which you want to export the virtual machine.
      After the export has completed, you will have a directory containing a number of files and subdirectories. Copy the entire directory from the Hyper-V server to a directory on your VDI server or to a shared directory on a remote server (the shared directory must be accessible to the VDI server).

2. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

3. Import the desktop into VDI.
   a. Open the Pools tab, then select the previously created empty pool.
   b. Select the Desktops tab, and click Import. An import dialog will be displayed.
   c. In the Server property, select the server you copied the Hyper-V desktop directories to (if you copied them to your VDI server

then choose the VDI Host '<servername>' option otherwise if you copied them to a shared directory on a remote server then choose the Other Server option and enter the remote server name where the shared directory resides).

    d. In the Path property, enter the path to the directory that contains the Hyper-V desktop directories.

    e. Select the correct desktop name from the Desktop dropdown, and click OK.

After the desktop has been imported successfully, it will display in the Desktops tab of the Pools page. A page refresh might be necessary.

# How to Clone Desktops (Sun VirtualBox)

Cloning is the fastest and most efficient way to populate a pool. Use the steps below to enable cloning in a pool.

### Before You Begin

A virtual machine must be imported before a template can be cloned. Refer to the How to Clone Desktops (Sun VirtualBox) page for detailed information.

### VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Enable cloning in a desktop pool.
   - If you would like to enable cloning in an existing pool, it can be done on the pool's Cloning tab.
     a. In the VDI Manager, open the Pools tab, then select the previously created pool.
     b. Select the Cloning tab, and specify the cloning parameters.
     c. At a minimum, define a desktop template to clone from, and select Enable Automatic Cloning.
   - Alternatively, you can enable cloning during pool creation in the New Pool wizard.
     a. Choose the desktop template and select Enable Automatic Cloning.
     b. Click Finish to finalize the pool creation, and begin the automatic cloning.

Cloning can take up to a minute to start, after which you will see clone jobs begin to display in the Jobs window. To access the Jobs window, click the Jobs Running link in the top left of the VDI Manager. After a clone job has been finished successfully, the new desktop will display in the Desktops tab of the Pool page. A page refresh might be necessary.

### CLI Steps

1. Open a terminal window and sign into the server with root credentials.
   For a multi-host configuration, use one of the VDI Secondary hosts.

2. Start automatic cloning in a pool.

```
# /opt/SUNWvda/sbin/vda pool-start <pool name>
```

- Example – Starting automatic cloning in a pool

```
example% /opt/SUNWvda/sbin/vda pool-start MyPool
```

# How to Clone Desktops (VMware vCenter)

Cloning is the fastest and most efficient way to populate a pool. Use the steps below to enable cloning in a pool.

### Before You Begin

A virtual machine must be imported before a template can be cloned. Refer to the How to Clone Desktops (VMware vCenter) page for detailed information.

### VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Pools category, then select a pool's Resources tab.

3. Select your preferred storage for newly cloned virtual machines.
   By default, all available storage may be used. For each clone, the VDI Core will select the storage with the most available disk space.

4. Select the Desktop Cloning tab.

5. Select the preferred template from the Template menu.
   The menu will list all templates that are available in the VMware vCenter.

6. Select Apply System Preparation, and specify which Customization Specification should be used.

7. Select Enable Automatic Cloning, and click Save.

Cloning can take up to a minute to start, after which you will see clone jobs begin to display in the Jobs window. To access the Jobs window, click the Jobs Running link in the top left of the VDI Manager. After a clone job has been finished successfully, the new desktop will display in the Desktops tab of the Pool page. A page refresh might be necessary.

### CLI Steps

1. Open a terminal window and sign into the server with root credentials.
   For a multi-host configuration, use one of the VDI Secondary hosts.

2. Start automatic cloning in a pool.

```
# /opt/SUNWvda/sbin/vda pool-start <pool name>
```

- Example – Starting automatic cloning in a pool

```
example% /opt/SUNWvda/sbin/vda pool-start MyPool
```

# How to Clone Desktops (Microsoft Hyper-V)

Cloning is the fastest and most efficient way to populate a pool. Use the steps below to enable cloning in a pool.

### Before You Begin

A virtual machine must be imported before a template can be cloned. Refer to the How to Clone Desktops (Microsoft Hyper-V) page for detailed information.

### VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Enable cloning in a desktop pool.
   - If you would like to enable cloning in an existing pool, it can be done on the pool's Cloning tab.
     a. In the VDI Manager, open the Pools tab, then select the previously created pool.
     b. Select the Cloning tab, and specify the cloning parameters.
     c. At a minimum, define a desktop template to clone from, and select Enable Automatic Cloning.
   - Alternatively, you can enable cloning during pool creation in the New Pool wizard.
     a. Choose the desktop template and select Enable Automatic Cloning.
     b. Click Finish to finalize the pool creation, and begin the automatic cloning.

Cloning can take up to a minute to start, after which you will see clone jobs beginning to display in the Jobs window. To access the Jobs window, click the Jobs Running link in the top left of the VDI Manager. After a clone job has been finished successfully, the new desktop will display in the Desktops tab of the Pool page. A page refresh might be necessary.

### CLI Steps

1. Open a terminal window and sign into the server with root credentials.
   For a multi-host configuration, use one of the VDI Secondary hosts.

2. Start automatic cloning in a pool.

```
# /opt/SUNWvda/sbin/vda pool-start <pool name>
```

- Example – Starting automatic cloning in a pool

```
example% /opt/SUNWvda/sbin/vda pool-start MyPool
```

# About Template Management

In Sun VDI 3, making changes to imported desktops was cumbersome. You either needed to make adaptations outside of Sun VDI and re-import the desktop image again, or you were forced to temporarily assign the desktop to a user, access the desktop on behalf of this user, and then make your changes. Sun VDI 3.1 introduces an Adobe Flash plug-in that enables you to easily access, test, and modify the desktop directly from within the VDI Manager. This feature also includes changing desktop properties as well as mounting ISO images for setting up the operating system.

Any desktop can be used as template for cloning additional desktops. Testing desktop templates and keeping track of any changes before rollout is crucial for large enterprise deployments. Sun VDI now includes support for managing several template revisions. You can create a new template revision at any time, test your changes and declare the new revision as the master used for the cloning process. You can also revert to a previous revision if you are not satisfied with your changes.

Sun VDI leverages the Microsoft System Preparation tool (Sysprep) for preparing Windows desktops for cloning. The use of Sysprep ensures that each desktop clone is assigned its own unique security identifier (SID), which is mandatory if desktops need to join an Active Directory domain. In Sun VDI 3.1 it is now possible to trigger Sysprep from within the VDI Manager. The corresponding template revision is automatically marked as Sysprepped once the preparation has completed.

# How to Create and Modify a Desktop Template in the VDI Manager

The Template Management is available for Sun VirtualBox and Microsoft Hyper-V desktop pools. This functionality is especially useful for installation of additional software or operating system upgrades.

### Before You Begin

You will need to have created at a minimum a virtual machine in the interface of your chosen desktop provider (Sun VirtualBox, or Microsoft Hyper-V) before you can import it and use the template modification tools in the VDI Manager. Once you have created and imported the virtual machine, you can start it from the VDI Manager and carry out all the necessary preparation steps from there. For more information, see How to Create Virtual Machines (Sun VirtualBox) or How to Create Virtual Machines (Microsoft Hyper-V).

VDI Manager Steps

1. Sign into the VDI Manager.
    a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
    b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Click the pool that you would like to populate with desktops, then select the Template tab.

3. Click the Import Template button to import the bare virtual machine you just created in the desktop provider interface.
   The virtual machine will be saved as Revision 1 as soon as it is imported.

4. Select the template you would like to modify, and click Start from the More Actions menu.
   No modifications can be performed until the virtual machine is started from the VDI Manager.

5. If necessary, make modifications to the template, such as installation of additional software or upgrades of the operating system.
    - Virtual machines hosted by Sun VirtualBox can be modified from an interactive Adobe Flash console.

      

    - On Microsoft Hyper-V desktop providers, the modifications may take place on the desktop provider's Hyper-V host.

6. When you are finished modifying the template, select Shut Down from the More Actions menu.

7. Apply template revisions.
    - Create a new revision - Click the Create Revision button.
    - Revert back to a previous revision - Select Revert from the More Actions menu.
    - Approve a revision for the template - Click the Make Master button.
    - Populate the pool with clones of the template's Master (or approved) Revision - Chose a template and select Apply To Pool from More Actions
    - Populate the pool with clones of a revision, not necessarily approved - Chose a revision and select Apply To Pool from More Actions
    - Copy a revision to a new template - Select Copy To Template in More Actions.
    - Delete a revision or template - Select the revision or template and click Delete from More Actions.
      If a template is deleted, all its revisions and all desktops cloned from the template will be deleted too.

# How to Enable System Preparation for Windows Templates (VirtualBox and Hyper-V)

Windows desktops require System Preparation for successful cloning by VDI. After you create a Windows virtual machine, you should prepare it for Sysprep by downloading a Sysprep CAB (Windows XP only), and installing the VDA Tools (Hyper-V virtualization platforms only). Import the virtual machine into the VDI Core as a template, and select System Preparation on one of the template revisions. The VDI Core boots the revision, runs Sysprep.exe, and then shuts down the system. The revision now acts as a blank slate for cloning desktops in any pool with a valid System Preparation file.

A pool's System Preparation file defines licensing and credentials. If a pool has a valid System Preparation file, System Preparation is enabled, and cloning from the sysprepped template is enabled, all cloned desktops in the pool will have the customization defined by the System Preparation file.

One sysprepped revision can be used for multiple pools, and the System Preparation files can be changed and saved at any time from within the VDI Manager.

> ℹ️ Due to a bug in Windows 7, the Windows Media Player Network Sharing Service causes the Windows Sysprep tool to hang. If you do not need this service enabled in your Windows 7 desktops and you intend to run System Preparation from VDI, stop and disable it. If you prefer to leave this service enabled, run Sysprep manually from within the template before importing it.

Before You Begin

1. (Hyper-V Only) Install the VDA Tools on the template.
   The System Preparation action in the Template tab will not work if you do not have the tools (`vda-tools.msi`) installed on your template. For Windows XP templates, you also need to have the Sysprep tools in a `C:\Sysprep` directory.

2. Prepare the template for System Preparation.
   - Windows XP
     a. Log into the template and download the appropriate Sysprep CAB for your version of Windows XP.
        - Windows XP Service Pack 2 Deployment Tools
        - Windows XP Service Pack 3 Deployment Tools
     b. Create a directory on the template named `C:\Sysprep`.
     c. Unpack the contents of the Sysprep CAB into the `C:\Sysprep` directory.
   - Windows Vista & Windows 7
     No files need to be installed. Windows Vista and Windows 7 ship with all required system preparation files pre-installed.

3. Import a virtual machine template in the VDI Manager.
   For more information, refer to one of the following pages:
   - How to Import Desktops (Sun VirtualBox)
   - How to Import Desktops (Microsoft Hyper-V)

VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Run System Preparation in a template revision.
   a. Click the Template tab, and select a revision.
   b. Choose System Preparation from the More Actions menu.
      This action will start a job, start the revision, run `Sysprep.exe`, and wait for the system to shut down.
   c. Wait for the job to complete successfully via the Job Summary Pop-Up. If the job fails for any reason, details of the failure can viewed in the Job Details text area by clicking on the failed job.
   d. Select the sysprepped revision and click Make Master.
      All pools currently using this template will clone new desktops from the sysprepped revision.

3. Prepare a pool to clone customized desktops based on a System Preparation file.
   a. Go to a pool's Cloning tab, or the Select Template screen of the New Pool wizard.
   b. Create a System Preparation file.
      The file requires a Windows administrator password, a Windows license key, and a Windows workgroup or a Windows domain, domain administrator, and administrator password.
   c. Select the sysprepped template, and select Apply System Preparation.
      You are now ready to clone customized Windows desktops. Refer to the How to Clone Desktops (Sun VirtualBox) and How to Clone Desktops (Microsoft Hyper-V) pages.

# How to Create Automated Administration Scripts

The `/opt/SUNWvda/sbin/vda` CLI can be used in scripts for automated administration.

## Reading the Return Code

The `/opt/SUNWvda/sbin/vda` returns the following exit codes:

- 0: Successful completion
- 1: An error occurred
- 2: Invalid command line options or arguments were specified

## Waiting for a Job to Finish

Some `vda` subcommands return immediately but start an action in the background, a job.
The subcommand `job-wait` allows to synchronously wait for a specific job to be completed.

```
# /opt/SUNWvda/sbin/vda job-wait --help
Wait until the job ends

Usage:
 vda job-wait [-t <timeout> | --timeout=<timeout>] <job>
 -?, --help                Print this help list
Options:
 -t <timeout>, --timeout=<timeout>
                           Timeout in seconds to wait
Operand:
*<job>                     The id of the job
'*' denotes mandatory parameters.
```

## Parsing the Output of the CLI

A number of subcommands support a parseable option so that the output is formatted for easy parsing: as a list of lines of colon-separated (':') fields.

The syntax of the option is:

```
 -x, --parseable         Display output suitable for programmatic parsing.
```

### user-search

Search for users/groups in the user directory that match the specified search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Name of the user/group | string |
| Kind of object | User / Group |
| DN of the user/group | string |

### user-show

Show the desktops available for the user.

Parseable Output in the case of a user: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | User / Token <token> / Group <group_name> / Custom Group <group_name> |

Parseable Output in the case of a group: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |

### user-desktops

Show the desktops assigned to the user.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | flexible / personal |
| Is Default Desktop | true / false |

### group-list

Lists all custom groups.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Custom Group Name | string |

### group-show

Show the pools assigned to the custom group.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |

### token-search

Search for tokens that match the search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Token | string |
| Name of the Associated User | string |
| DN of the Associated User | string |

### token-show

Show the desktops available for the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|

| | |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | User / Token / Group <group_name> / Custom Group <group_name> |

### token-desktops

Show the desktops assigned to the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | flexible / personal |
| Is Default Desktop | true / false |

### pool-list

List all pools.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Type of Desktop Assignment | Personal / Flexible |
| Number of Desktops | integer |
| Desktop Provider Name | string |

### pool-show

Show detailed information about the pool.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Assignment Status | Enabled / Disabled |
| Type of Desktop Assignment | Personal / Flexible |
| Desktop Provider Name | string |
| Cloning Status | Enabled / Disabled |
| Template | None / string |
| Number of Cloning Jobs | integer |
| Number of Available Desktops | integer |
| Number of Assigned Desktops | integer |
| Total Number of Desktops | integer |

## pool-desktops

List all desktops from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
| --- | --- |
| Desktop Name | string |
| Desktop ID | long |
| Machine State | `Running` / `Powered Off` / `Suspended` / `Unknown` |
| Desktop State | `Available` / `Used` / `Idle` / `Unresponsive` / `Reserved` / etc. |
| DN of Assigned User | string |

## pool-templates

List all templates from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
| --- | --- |
| Template Name | string |
| Template ID | long |
| Machine State | `Running` / `Powered Off` / `Suspended` / `Aborted` / `Unknown` |
| Master Revision | string |
| Cloned Desktops | string |

## template-revisions

List the revisions of the template.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
| --- | --- |
| Revision Name | string |
| Revision ID | long |
| Creation Date | timestamp |
| Is It Master | `yes` / `no` |
| Cloned Desktops | string |

## provider-list

List all desktop providers.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
| --- | --- |
| Desktop Provider Name | string |
| Provider Type | `Sun VirtualBox` / `VMware vCenter` / `Microsoft Hyper-V` / `Microsoft Remote Desktop` |
| Total Number of Desktops | integer |

| | |
|---|---|
| Number of Used Desktops | integer |
| CPU Usage | `xx% (x.x GHz/MHz)` |
| Memory Usage | `xx% (x.x GB/MB)` |
| Storage Usage | `xx% (x.x GB/MB)` |

### provider-list-hosts

List all hosts for the VirtualBox desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Host Name | string |
| Status | `OK` / `Unresponsive` / etc. |
| Enabled | `Enabled` / `Disabled` |
| CPU Usage | `xx% (x.x GHz/MHz)` |
| Memory Usage | `xx% (x.x GB/MB)` |
| Number of Desktops | integer |

### provider-list-storage

List all storage servers for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Status | `OK` / `Unresponsive` / etc. |
| Enabled | `Enabled` / `Disabled` |
| ZFS Pool | string |
| Capacity | `xxx.x GB` |
| Usage | `xx.x GB` |
| Number of Desktops | integer |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Storage ID | string |
| ZFS Pool | string |
| Capacity | `xxx.x GB` |
| Usage | `xx.x GB` |
| Number of Desktops | integer |

### provider-list-templates

List the templates for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|-------|-------------|
| Template Name | string |
| Template ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|-------|-------------|
| Template Name | string |
| Template ID | string |
| Path | string |

## provider-list-unmanaged

List the desktops from the virtualization platform that are not managed by any desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|-------|-------------|
| Host Name | string |
| Desktop Name | string |
| Desktop ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|-------|-------------|
| Desktop Name | string |
| Desktop ID | string |

## provider-list-networks

List all networks for the desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|-------|-------------|
| Subnet Label | String |
| Subnet Address | String |
| Availability | `All Hosts` / `Not on: <comma_separated_list_of_hosts>` |

## job-list

List the existing jobs.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|-------|-------------|

| | |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| ID of the Job | integer |
| Cancellable | 'C' if the job can be cancelled |

## job-show

Show the job details.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| Start Time | hh:mm:ss |
| End Time | hh:mm:ss |
| Job Details | string |
| Cancellable | `true` / `false` |

## Contents

# User Directory Management (All Topics)

## About User Directory Integration

Typically user information is already stored in an Active Directory or LDAP server. Before you can assign users to desktops, you must configure the desired Active Directory/LDAP server and the VDI Core. The following information describes the user directory types supported by VDI.

### Active Directory Types

Active Directory integration is the recommended choice for production platforms integrating with Microsoft Active Directory. Active Directory integration requires additional configuration (Kerberos configuration and time synchronization) on the VDI host. If you just want to quickly set up a demo with an Active Directory, it should be more straight-forward to use LDAP Types.
The users from the Active Directory can be used for desktop and pool assignments and will be able to access desktops from VDI. On top of this basic feature, Active Directory integration offers the following functionalities:

1. Active Directory integration allows to access all the users from a forest and use those users for desktop and pool assignments.
   This means that the users from the different sub-domains of the forest will be able to access desktops from VDI.
2. Active Directory integration allows computer entries to be removed from the Active Directory when cloned desktops are deleted by the VDI Core.
   When a Windows desktop (cloned by VDI) joins a domain (through Sysprep), this will typically create a new computer entry in the Active Directory. Configuring VDI with Kerberos Authentication will allow VDI to remove the computer entries from the Active Directory, when deleting unused desktops. This avoids having computer entries piling up in the Active Directory while the matching desktops have long been destroyed.
3. Active Directory integration allows end-users to update their password in the Active Directory server either before this password has expired (optional action) or after the password has expired (mandatory action).

You can choose from the following VDI-supported Active Directory types:

- Kerberos Authentication - The typical choice when integrating with Microsoft Active Directory.
  See the How to Set Up Kerberos Authentication page for more information.

- Public Key Authentication - To be used to integrate with Microsoft Active Directory when the domain controller requires LDAP signing.
  See the How to Set Up Public Key Authentication page for more information.

### LDAP Types

LDAP integration is the recommended choice for integrating with other types of LDAP directories or to quickly set up a demo with Active Directory. The setup is straight-forward, without the need for extra configuration.
If you need to install your own directory, you may choose OpenDS. Some directions to set it up for VDI can be found here.

> ⓘ  LDAP Integration allows end-users to update their password in the directory server only before this password has expired. If the user password expires, the end-user will be required to update her password using a customer-provided process external to Sun VDI.

LDAP Integration offers three security types for authentication: anonymous, simple, and secure:

- Anonymous Authentication - Useful for a quick integration with an LDAP server for demo purposes. Anonymous Authentication may only be chosen if your LDAP server supports anonymous authentication. It is not recommended to select Anonymous Authentication on production platforms. Active Directory does not support Anonymous Authentication.
  See the How to Set Up Anonymous Authentication page for more information.

- Simple Authentication - The demo solution for Active Directory and the typical choice for other LDAP directories. Simple Authentication is the recommended choice for production platforms integrating with LDAP directories other than Active Directory. If integrating with Active Directory, it is not recommended to select Simple Authentication on production platforms as a better integration can be achieved using Kerberos Authentication. A default restriction in Active Directory prevents password update from an LDAP Simple Authentication.

See the How to Set Up Simple Authentication page for more information.

- Secure Authentication - Useful to secure connections over SSL, when the directory supports it.
  See the How to Set Up Secure Authentication page for more information.

## User Directory Customization

If you have an expert understanding of user directory integration and would like to optimize VDI for your user directory, please refer to the following pages:

- About Internal Usage of the LDAP Filters and Attributes
- How to Edit the LDAP Filters and Attributes
- How to Reconfigure the User Directory Settings

# How to Set Up Kerberos Authentication

Follow the steps below to configure Kerberos Authentication for your Active Directory.

> ℹ️ To get the full functionality offered by Kerberos Authentication, it is necessary to provide the credentials of a user that has 'write' access to the Active Directory. This user will be used to read users and delete computer entries from the directory.

### Steps

Kerberos Authentication requires some specific configuration on the Active Directory server and VDI host prior to setting up the user directory in the VDI Manager:

1. Kerberos authentication must be enabled in Active Directory.
   It should already be enabled as the default.

2. Ensure that each Active Directory domain has a global catalog server.
   Configure a domain controller in each domain as a global catalog server.

3. Synchronize the time between the VDI server and Active Directory server.
   For example, use `ntpdate` <my.windows.host>

4. Edit the system default Kerberos configuration file (**/etc/krb5/krb5.conf** on Solaris OS platforms) on the VDI server.
   At a minimum, the Kerberos configuration file must contain the following sections:
   - `[libdefaults]` - this sets defaults for Kerberos authentication. You must set the `default_realm` and `default_checksum`.
   - `[realms]` - this sets the KDCs for each Kerberos realm. A realm can have more than one `kdc`, the port can omitted if the default port 88 is used.
     To allow end-users to update their password, the details of the server that handles the password change for each Kerberos realm must be specified. The `kpasswd_server` and `admin_server` entries identify the Kerberos administration server that handles the password change. If `kpasswd_server` is omitted, the `admin_server` is used instead. The port can be omitted if the default port 464 is used.
     Format of a realm definition:

     ```
     <REALM_NAME> = {
     kdc = <host:port>
     kdc = <host:port>
     ...
     kpasswd_server = <host:port>
     admin_server = <host:port>
     kpasswd_protocol = SET_CHANGE
     }
     ```

   - `[domain_realm]` - this maps Active Directory domains to Kerberos realms.
     The following is an example Kerberos configuration file for a domain containing just one server:

```
[libdefaults]
default_realm = MY.COMPANY.COM
default_checksum = rsa-md5


[realms]
MY.COMPANY.COM = {
kdc = my.windows.host
admin_server = my.windows.host
kpasswd_protocol = SET_CHANGE
}

[domain_realm]
.my.company.com = MY.COMPANY.COM
my.company.com = MY.COMPANY.COM
```

5. You can check that Kerberos and its name resolution requirements are configured properly by using **getent**, **nslookup** and **kinit**. For example:
    - `# getent hosts my.windows.host` must return the IP address and the hostname
    - `# getent hosts <IP_of_my.windows.host>` must return the IP address and the hostname
    - `# nslookup -query=any _gc._tcp.my.company.com` must resolve the domain
    - `# kinit -V super-user@MY.COMPANY.COM` must succeed

6. Restart the Common Agent Container:

```
cacaoadm stop --force
cacaoadm start
```

7. In the VDI Manager, go to the Settings category and User Directory subcategory, and click Add User Directory to launch the User Directory wizard:
    a. Select Active Directory Type, and click Next.
    b. Select Kerberos Authentication.
    c. Enter the domain for the Active Directory
       For example: `my.company.com`
    d. Enter the user principal name of a user that has sufficient privileges to write into the Active Directory.
       For example: `super-user` or `super-user@my.company.com`
    e. Enter the password for that user.
    f. Click Next to review your choices before completing the configuration.

## How to Set Up Public Key Authentication

Public Key Authentication requires some specific configuration on the Active Directory server and VDI host prior to setting up the user directory in the Admin GUI.

### Steps

1. Follow the configuration steps 1 to 5 described for Kerberos Authentication
2. Creating a client certificate for each of the VDI host.
   The VDI keystore for the client certificate is located at `/etc/opt/SUNWvda/sslkeystore` and the password is `changeit`.
    a. Generating a key pair (private/public key) for the client certificate.
        - on the VDI host, log in as superuser (root) and use `keytool` to generate the key pair in the VDI keystore.

```
/usr/java/jre/bin/keytool -genkey -keyalg rsa \
-keystore /etc/opt/SUNWvda/sslkeystore \
-storepass changeit -keypass changeit \
-alias <your_alias>
```

b. Generating a Certificate Signing Request (CSR) for client certificate.
  - on the VDI host, use `keytool` to generate the certificate request.

```
/usr/java/jre/bin/keytool -certreq \
-keystore /etc/opt/SUNWvda/sslkeystore \
-storepass changeit -keypass  changeit \
-alias <your_alias> \
-file <certreq_file>
```

The alias must be the same as the alias used when generating the key pair. Aliases are case-insensitive.

c. Creating the certificate.
  - Copy the CSR file to the server hosting the Active Directory.
  - Using Internet Explorer, go to `"http://localhost/certsrv"`.
  - Log in.
  - On the Microsoft Certificate Services page, click Request a certificate.
  - On the Request a Certificate page, click advanced certificate request.
  - On the Advanced Certificate Request page, click Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file.
  - On the Submit a Certificate Request or Renewal Request page, paste the contents of the CSR into the Saved Request text box or browse to the CSR file.
  - Select an appropriate template from the Certificate Templates list. (Administrator is recommended).
  - Click Submit.
  - On the Certificate Issued page, ensure Base 64 encoded is selected and click Download certificate chain.
  - Save the certificate file.
d. Importing the certificate on the VDI host.
  - Copy the certificate file to the VDI host.
  - Import the certificate into the VDI keystore

```
/usr/java/jre/bin/keytool -import  \
-keystore /etc/opt/SUNWvda/sslkeystore \
-storepass changeit -keypass changeit \
-trustcacerts -file <certificate_file> \
-alias <your_alias>
```

3. Restart the Common Agent Container:

```
cacaoadm stop --force
cacaoadm start
```

4. Configuring the user directory in VDI Admin GUI.
   In the Admin GUI, go to the Settings category and User Directory subcategory, and click Add User Directory to launch the User Directory wizard:
   a. Select Active Directory Type, and click Next.
   b. Select Public Key Authentication.
   c. Enter the domain for the Active Directory.
      For example: `my.company.com`
   d. The following step shows the SSL certificates of the Active Directory servers. Click Next to permanently accept the certificates.
   e. Click Next to review your choices before completing the configuration.


## How to Set Up Anonymous Authentication

Use the steps below to set up anonymous authentication.

Steps

In the Admin GUI, go to the Settings category and User Directory subcategory, and click Add User Directory to launch the User Directory wizard:

1. Select LDAP Type, and click Next.

2. Select Anonymous Authentication.

3. Enter the hostname or IP address, and port number, of the LDAP server.
   The default port number, 389, is used by most LDAP servers.

4. Enter the base DN of the LDAP server. Specifying a base DN is optional.
   It allows you to restrict the part of the LDAP directory used to search for users.
   For example: `cn=Users,dc=my,dc=company,dc=com`

5. Click Next to review your choices before completing the configuration.


## How to Set Up Simple Authentication

Use the steps below to set up simple authentication.

> ℹ️ It is necessary to provide the credentials of a user that has 'read' access to the user directory. This user will be used to read user information from the directory.

Steps

In the Admin GUI, go to the Settings category and User Directory subcategory, and click Add User Directory to launch the User Directory wizard:

1. Select LDAP Type, and click Next.

2. Select Simple Authentication.

3. Enter the hostname or IP address, and port number, of the LDAP server.
   The default port number,389, is used by most LDAP servers.

4. Enter the base DN of the LDAP server. Specifying a base DN is optional.
   It allows you to restrict the part of the LDAP directory used to search for users.
   For example: `cn=Users,dc=my,dc=company,dc=com`

5. Enter the user name.
   It must be the fully distinguished name (DN) of a user that has sufficient privileges to search the LDAP directory.
   For example: `cn=super-user,cn=Users,dc=my,dc=company,dc=com`.

6. Enter the password for the user.

7. Click Next to review your choices before completing the configuration.


## How to Set Up Secure Authentication

Use the steps below to set up secure authentication.

> ℹ️ It is necessary to provide the credentials of a user that has 'read' access to the user directory. This user will be used to read user information from the directory.

Steps

In the Admin GUI, go to the Settings category and User Directory subcategory, and click Add User Directory to launch the User Directory wizard:

1. Select LDAP Type, and click Next.

2. Select Secure Authentication.

3. Enter the hostname or IP address, and port number, of the LDAP server.
   The default port, 636, is used by most SSL secured LDAP servers.

4. Enter the base DN of the LDAP server.
   Specifying a base DN is optional. It allows you to restrict the part of the LDAP directory used to search for users.
   For example: `cn=Users,dc=my,dc=company,dc=com`

5. Enter the user name.
   It must be the fully distinguished name (DN) of a user that has sufficient privileges to search the LDAP directory.
   For example: `cn=super-user,cn=Users,dc=my,dc=company,dc=com`.

6. Enter the password for the user.

7. The following step shows the SSL certificate of the LDAP server.
   Click Next to permanently accept the certificate.

8. Review your choices before completing the configuration.


# About Internal Usage of the LDAP Filters and Attributes

VDI uses various LDAP filters and attribute lists to look up and interpret the data stored in the user directory.

This sections explains how the LDAP filters and attributes are used by Sun VDI to perform the different searches in the user directory necessary for each task.
See How to Edit the LDAP Filters and Attributes for details about how to edit those filters.

### Searching for Users and Groups

You can use the administration tools (VDI Manager or CLI) to search for users and groups, in order to assign them to desktops or pools.

The search logic works as follow:

- Users are searched first:
  - the filter used to search for users is: `(&<ldap.user.object.filter><ldap.user.search.filter>)`.
  - the `$SEARCH_STRING` placeholder is replaced by `*criteria*` where criteria is the string typed in the VDI Manager search field. If the criteria string already contains a wild-card `*`, then the `$SEARCH_STRING` placeholder is simply replaced by `criteria`.
- Groups are then searched as follow:
  - the filter used to search for users is: `(&<ldap.group.object.filter><ldap.group.search.filter>)`.
  - the `$SEARCH_STRING` placeholder is replaced by `*criteria*` where criteria is the string typed in the VDI Manager search field. If the criteria string already contains a wild-card `*`, then the `$SEARCH_STRING` placeholder is simply replaced by `criteria`.

> ℹ️ If the global setting `ldap.search.wildcard` is set to disabled, the `$SEARCH_STRING` placeholder is replaced by `criteria` (without being surounded by the wildcards). This restricts the returned results to strictly match the typed string but it is useful with very large and distributed user directories where the search using the wildcards takes too long to return. Wildcards are added by default as the default value for `ldap.search.wildcard` is enabled.

### Requesting a Desktop for a User

When requesting a desktop for a user, VDI first needs to find the user DN that matches the user ID before resolving the pool/desktop assignments for the user DN. If client authentication is enabled, then the user ID attribute is also used for authentication.

The attributes used to match the user ID are defined in `ldap.userid.attributes`.

### Resolving Group Membership

Group membership is resolved using the attributes defined in `ldap.user.member.attributes` and `ldap.group.member.attributes`. Nested group depth is limited to 3.

VDI also resolves Primary Group membership which is Active Directory specific. The attributes used for resolving primary group membership are defined in `ldap.group.short.attributes` and `ldap.user.member.attributes`.

### LDAP Cache

In order to improve the performance and reduce the load on the user directory, the user and group entries retrieved by VDI are cached. Entries in the LDAP cache time out after 10 minutes.

It is not possible, at the moment, to change the LDAP cache timeout, nor to flush the cache.

# How to Edit the LDAP Filters and Attributes

VDI uses various LDAP filters and attribute lists to look up and interpret the data stored in the user directory.

VDI comes with some default LDAP filters that are suitable for demos with Active Directory or Sun Directory Server. But these filters might be incompatible with other types of directories such as OpenLDAP or eDirectory and would then need to be modified.

For production, it is always recommended to customize those filters to match most closely the LDAP schema of the directory.

This section explains how to edit those filters, and the values recommended per type of directory.
See About Internal Usage of the LDAP Filters and Attributes for details about how Sun VDI makes use of the different filters and attributes.

### Before You Begin

The LDAP filters are specified as global settings using the `vda` command:

- /opt/SUNWvda/sbin/vda settings-getprops
- /opt/SUNWvda/sbin/vda settings-setprops

We recommend reviewing the command parameters before editing the LDAP filters.

### CLI Steps

1. List the LDAP filter used to identify objects of type 'user' and the LDAP filter used to search for users according a search criteria:

```
example% /opt/SUNWvda/sbin/vda settings-getprops -p
ldap.user.object.filter,ldap.user.search.filter
ldap.user.object.filter:

(&(|(objectclass=user)(objectclass=person)(objectclass=inetOrgPerson)(objectclass=organization

ldap.user.search.filter:
                        (|(cn=$SEARCH_STRING)(uid=$SEARCH_STRING)(mail=$SEARCH_STRING))
```

2. Customize the LDAP filter used to search for users according a search criteria, for Active Directory:

```
example% /opt/SUNWvda/sbin/vda settings-setprops -p
ldap.user.search.filter='"(|(cn=\$SEARCH_STRING)(uid=\$SEARCH_STRING)(mail=\$SEARCH_STRING))"'
updated.

example% /opt/SUNWvda/sbin/vda settings-getprops -p ldap.user.search.filter
ldap.user.search.filter:
                        (|(cn=$SEARCH_STRING)(uid=$SEARCH_STRING)(mail=$SEARCH_STRING))
```

# Default LDAP Filters and Attributes

| Global Setting Name | Description | Default Value |
| --- | --- | --- |
| ldap.user.object.filter | LDAP filter used to identify objects of type user | (&(\|(objectclass=user)(objectclass=person)(objectclass=inet (objectclass=organizationalPerson))(!(objectclass=computer) |
| ldap.user.search.filter | LDAP filter used to search for users according a search criteria. Searches for users can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | (\|(cn=$SEARCH_STRING)(uid=$SEARCH_STRING)(mail=$SEARCH_STRI |
| ldap.userid.attributes | List of comma separated LDAP attributes storing the userid value for user objects. This is used to find a user given its userid | uid,sAMAccountName |
| ldap.user.member.attributes | List of comma separated LDAP attributes on a user object storing the groups the user is a member of | memberof,primaryGroupID |
| ldap.group.object.filter | LDAP filter used to identify objects of type group | (\|(objectclass=group)(objectclass=groupofnames)(objectclass |
| ldap.group.search.filter | LDAP filter used to search for groups according a search criteria. Searches for groups can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | (\|(dc=$SEARCH_STRING)(o=$SEARCH_STRING)(ou=$SEARCH_STRING) (cn=$SEARCH_STRING)(uid=$SEARCH_STRING)(mail=$SEARCH_STRING |

| | | |
|---|---|---|
| `ldap.group.member.attributes` | List of comma separated LDAP attributes on a group object storing the users member of the group | `member,uniquemember` |
| `ldap.group.short.attributes` | List of comma separated LDAP attributes on a group object storing the information for primary group membership. Primary group membership is specific to Active Directory. | `primaryGroupToken` |
| `ldap.container.object.filter` | LDAP filter used to identify objects of type container. Containers can be selected as root for custom group filters in the web administration console | `(|(objectclass=domain)(objectclass=organization)(objectclass=organizationalUnit)(objectclass=container))` |
| `ldap.container.search.filter` | LDAP filter used by the web administration console to search for containers according a search criteria, when selecting a root for a custom group filter. $SEARCH_STRING is the place holder for the search criteria | `(|(cn=$SEARCH_STRING)(dc=$SEARCH_STRING)(ou=$SEARCH_STRING)` |
| `ldap.default.attributes` | List of comma separated LDAP attributes loaded in the cache when looking up an object. It should contain all the attributes used in the other filters and attribute lists. | `dc,o,ou,cn,uid,mail,member,uniquemember,memberof,sAMAccountName,primaryGroupToken,primaryGroupID` |

## Recommended Values with Active Directory

| Global Setting Name | Description | Recommended Value with Active Directory |
|---|---|---|
| | | |

| `ldap.user.object.filter` | LDAP filter used to identify objects of type user | `(&(objectclass=user)(!(objectclass=computer)))` |
|---|---|---|
| `ldap.user.search.filter` | LDAP filter used to search for users according a search criteria. Searches for users can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(|(cn=$SEARCH_STRING)(sAMAccountName=$SEARCH_STRING))` |
| `ldap.userid.attributes` | List of comma separated LDAP attributes storing the userid value for user objects. This is used to find a user given its userid | `sAMAccountName` |
| `ldap.user.member.attributes` | List of comma separated LDAP attributes on a user object storing the groups the user is a member of | `memberof,primaryGroupID` |
| `ldap.group.object.filter` | LDAP filter used to identify objects of type group | `(objectclass=group)` |
| `ldap.group.search.filter` | LDAP filter used to search for groups according a search criteria. Searches for groups can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(cn=$SEARCH_STRING)` |
| `ldap.group.member.attributes` | List of comma separated LDAP attributes on a group object storing the users member of the group | `member` |

| | | |
|---|---|---|
| `ldap.group.short.attributes` | List of comma separated LDAP attributes on a group object storing the information for primary group membership. Primary group membership is specific to Active Directory. | `primaryGroupToken` |
| `ldap.container.object.filter` | LDAP filter used to identify objects of type container. Containers can be selected as root for custom group filters in the web administration console | `(objectclass=container)` |
| `ldap.container.search.filter` | LDAP filter used by the web administration console to search for containers according a search criteria, when selecting a root for a custom group filter. $SEARCH_STRING is the place holder for the search criteria | `(cn=$SEARCH_STRING)` |
| `ldap.default.attributes` | List of comma separated LDAP attributes loaded in the cache when looking up an object. It should contain all the attributes used in the other filters and attribute lists. | `cn,member,memberof,sAMAccountName,primaryGroupToken,primary` |

> ℹ️  If you require to use the `userPrincipalName` attribute instead of `sAMAccountName` for user identification, replace 'sAMAccountName' by 'userPrincipalName' in the above values as described in this page.

## Recommended Values with Sun Directory Server

| Global Setting Name | Description | Recommended Value with Sun Directory Server |
|---|---|---|
| `ldap.user.object.filter` | LDAP filter used to identify objects of type user | `(objectclass=person)` |

| | | |
|---|---|---|
| `ldap.user.search.filter` | LDAP filter used to search for users according a search criteria. Searches for users can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(|(cn=$SEARCH_STRING)(uid=$SEARCH_STRING))` |
| `ldap.userid.attributes` | List of comma separated LDAP attributes storing the userid value for user objects. This is used to find a user given its userid | `uid` |
| `ldap.user.member.attributes` | List of comma separated LDAP attributes on a user object storing the groups the user is a member of | `memberof` |
| `ldap.group.object.filter` | LDAP filter used to identify objects of type group | `(objectclass=groupofuniquenames)` |
| `ldap.group.search.filter` | LDAP filter used to search for groups according a search criteria. Searches for groups can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(cn=$SEARCH_STRING)` |
| `ldap.group.member.attributes` | List of comma separated LDAP attributes on a group object storing the users member of the group | `uniquemember` |
| `ldap.group.short.attributes` | List of comma separated LDAP attributes on a group object storing the information for primary group membership. Primary group membership is specific to Active Directory. | empty |

| | | |
|---|---|---|
| `ldap.container.object.filter` | LDAP filter used to identify objects of type container. Containers can be selected as root for custom group filters in the web administration console | `(|(objectclass=domain)(objectclass=organizationalUnit))` |
| `ldap.container.search.filter` | LDAP filter used by the web administration console to search for containers according a search criteria, when selecting a root for a custom group filter. $SEARCH_STRING is the place holder for the search criteria | `(|(dc=$SEARCH_STRING)(ou=$SEARCH_STRING))` |
| `ldap.default.attributes` | List of comma separated LDAP attributes loaded in the cache when looking up an object. It should contain all the attributes used in the other filters and attribute lists. | `dc,ou,cn,uid,uniquemember,memberof` |

## Recommended Values with OpenDS

| Global Setting Name | Description | Recommended Value with OpenDS |
|---|---|---|
| `ldap.user.object.filter` | LDAP filter used to identify objects of type user | `(objectclass=person)` |
| `ldap.user.search.filter` | LDAP filter used to search for users according a search criteria. Searches for users can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(|(cn=$SEARCH_STRING)(uid=$SEARCH_STRING))` |
| `ldap.userid.attributes` | List of comma separated LDAP attributes storing the userid value for user objects. This is used to find a user given its userid | `uid` |

| | | |
|---|---|---|
| `ldap.user.member.attributes` | List of comma separated LDAP attributes on a user object storing the groups the user is a member of | `memberof` |
| `ldap.group.object.filter` | LDAP filter used to identify objects of type group | `(objectclass=groupofuniquenames)` |
| `ldap.group.search.filter` | LDAP filter used to search for groups according a search criteria. Searches for groups can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(cn=$SEARCH_STRING)` |
| `ldap.group.member.attributes` | List of comma separated LDAP attributes on a group object storing the users member of the group | `uniquemember` |
| `ldap.group.short.attributes` | List of comma separated LDAP attributes on a group object storing the information for primary group membership. Primary group membership is specific to Active Directory. | empty |
| `ldap.container.object.filter` | LDAP filter used to identify objects of type container. Containers can be selected as root for custom group filters in the web administration console | `(\|(objectclass=domain)(objectclass=organizationalUnit))` |
| `ldap.container.search.filter` | LDAP filter used by the web administration console to search for containers according a search criteria, when selecting a root for a custom group filter. $SEARCH_STRING is the place holder for the search criteria | `(\|(dc=$SEARCH_STRING)(ou=$SEARCH_STRING))` |

| | | |
|---|---|---|
| `ldap.default.attributes` | List of comma separated LDAP attributes loaded in the cache when looking up an object. It should contain all the attributes used in the other filters and attribute lists. | `dc,ou,cn,uid,uniquemember,memberof` |

## Recommended Values with Open LDAP

| Global Setting Name | Description | Recommended Value with Open LDAP |
|---|---|---|
| `ldap.user.object.filter` | LDAP filter used to identify objects of type user | It is mandatory to remove `(!(objectclass=computer))` from the default filter. Recommended is `(objectclass=person)` |
| `ldap.user.search.filter` | LDAP filter used to search for users according a search criteria. Searches for users can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(|(cn=$SEARCH_STRING)(uid=$SEARCH_STRING))` |
| `ldap.userid.attributes` | List of comma separated LDAP attributes storing the userid value for user objects. This is used to find a user given its userid | `uid` |
| `ldap.user.member.attributes` | List of comma separated LDAP attributes on a user object storing the groups the user is a member of | `memberof` |
| `ldap.group.object.filter` | LDAP filter used to identify objects of type group | `(objectclass=groupofnames)` |
| `ldap.group.search.filter` | LDAP filter used to search for groups according a search criteria. Searches for groups can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(cn=$SEARCH_STRING)` |
| `ldap.group.member.attributes` | List of comma separated LDAP attributes on a group object storing the users member of the group | `member` |
| `ldap.group.short.attributes` | List of comma separated LDAP attributes on a group object storing the information for primary group membership. Primary group membership is specific to Active Directory. | empty |
| `ldap.container.object.filter` | LDAP filter used to identify objects of type container. Containers can be selected as root for custom group filters in the web administration console | ??? |

| | | |
|---|---|---|
| `ldap.container.search.filter` | LDAP filter used by the web administration console to search for containers according a search criteria, when selecting a root for a custom group filter. $SEARCH_STRING is the place holder for the search criteria | ??? |
| `ldap.default.attributes` | List of comma separated LDAP attributes loaded in the cache when looking up an object. It should contain all the attributes used in the other filters and attribute lists. | `cn,uid,member,memberof` |

## Recommended Values with Novell eDirectory

| Global Setting Name | Description | Recommended Value with Novell eDirectory |
|---|---|---|
| `ldap.user.object.filter` | LDAP filter used to identify objects of type user | It is mandatory to remove `(!(objectclass=computer))` from the default filt `(objectclass=user)` |
| `ldap.user.search.filter` | LDAP filter used to search for users according a search criteria. Searches for users can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | `(|(cn=$SEARCH_STRING)(uid=$SEARCH_STRING)(givenName=$SEARCH` |
| `ldap.userid.attributes` | List of comma separated LDAP attributes storing the userid value for user objects. This is used to find a user given its userid | `givenName,cn,uid` |
| `ldap.user.member.attributes` | List of comma separated LDAP attributes on a user object storing the groups the user is a member of | `groupMembership` |
| `ldap.group.object.filter` | LDAP filter used to identify objects of type group | `(|(objectclass=group)(objectclass=groupofnames)(objectclass` |

| | | |
|---|---|---|
| `ldap.group.search.filter` | LDAP filter used to search for groups according a search criteria. Searches for groups can be done using the user-search command or in the web administration console. $SEARCH_STRING is the place holder for the search criteria | ??? |
| `ldap.group.member.attributes` | List of comma separated LDAP attributes on a group object storing the users member of the group | `member,uniquemember` |
| `ldap.group.short.attributes` | List of comma separated LDAP attributes on a group object storing the information for primary group membership. Primary group membership is specific to Active Directory. | empty |
| `ldap.container.object.filter` | LDAP filter used to identify objects of type container. Containers can be selected as root for custom group filters in the web administration console | `(objectclass=organizationalUnit)` |
| `ldap.container.search.filter` | LDAP filter used by the web administration console to search for containers according a search criteria, when selecting a root for a custom group filter. $SEARCH_STRING is the place holder for the search criteria | ??? |

| | | |
|---|---|---|
| `ldap.default.attributes` | List of comma separated LDAP attributes loaded in the cache when looking up an object. It should contain all the attributes used in the other filters and attribute lists. | `cn,uid,givenName,groupmembership,member,uniquemember` |

# How to Reconfigure the User Directory Settings

User directory settings are configured in the Admin Web UI in the Settings category and User Directory subcategory.

## Defining the User Directory

The instructions to define the user directory are described in About User Directory Integration.

Only one user directory can be defined.

## Changing the Security Level

It is possible to change the security level for the connections to the user directory:

1. go to the LDAP or Active Directory tab (depending on the user directory type)
2. click the Edit button for the Security Level, to launch the wizard
3. switch to another security level and modify the other settings if necessary, such as the port, the username and the password
4. click Next to review your choices before completing the configuration update.

It is only possible to switch to a security level within the same type of user directory, LDAP or Active Directory. If you want to switch between LDAP and Active Directory, you have to remove the user directory and add it again.

In the case of LDAP connection type, it is not possible change the security level if additional hosts have been defined (see Adding Fallback Hosts ).

## Changing the Credentials

When using Kerberos, Simple or Secure authentication, it is possible to update the credentials used for opening the connection to the user directory:

1. go to the LDAP or Active Directory tab (depending on the user directory type)
2. click the Edit button for the Security Level, to launch the wizard
3. edit the username and the password as necessary
4. click Next to review your choices before completing the configuration update.

## Updating the Server SSL Certificates

When using Public Key or Secure authentication, if the SSL certificate for the server has been changed, you need VDI to use the new certificate:

1. go to the LDAP or Active Directory tab (depending on the user directory type)
2. click the Edit button for the Security Level, to launch the wizard
3. do not change any of the existing settings if you only want to update the server certificates
4. the following step shows the SSL certificates of the servers. Click Next to permanently accept the certificates.
5. click Next to review your choices before completing the configuration update.

## Adding Fallback Hosts

When using the LDAP type of connection, it is possible to have additional LDAP hosts that would be used as a fallback in the case the connection to the main host is failing.

The additional LDAP hosts must be the replica of the main host. The connection to the fallback hosts will be open using the same security level, same port, same base DN and same credentials as for the main host.

The list of LDAP hosts can be found in the LDAP tab. Hosts can be added, removed and their order can be changed.

### Removing the User Directory

The button to remove the user directory can be found on the LDAP or Active Directory tab.

If some assignments have been registered using some data (users or groups) from the user directory, a warning will be poped up and a confirmation is asked. If you confirm, the user directory will be removed, but it will leave VDI in a broken state where the users will not be able to access to their desktop. If you add the settings to the same directory again, even using a different security level, the assignments will still be valid and the user will be able to access their desktop again.

## Contents

# Token and User Management (All Topics)

# How to Assign Users to Pools or Desktops

You can either assign a user to a specific desktop, or you can assign a user (or user group) to a desktop pool. If a user is assigned to a pool and requests a desktop, Sun VDI will automatically deliver any available desktop from the pool.

For Microsoft Remote Desktop providers, users cannot be directly assigned to desktops. Instead, users or groups are assigned to Remote Desktop Services pools.

### VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Users category.
   - To assign a user or a group, select the Users and Groups subcategory in the left sidebar.
     a. Search for users and groups in the user directory.
        You can specify user name or user ID.
     b. Select a user or group name, and then the Assignment tab in the corresponding profile.
     c. Select Add in the either the user's Assigned Desktops or Assigned Pools table, or the group's Assigned Pools table.
   - To assign a custom group, select the custom group name in the left sidebar.
     a. Select the Assignment tab in the custom group's profile.
     b. Select Add in the custom group's Assigned Pools table.

3. In the pop-up window, choose the pool or desktop to be assigned, and click OK.

You can always see which pools and desktops are associated with a user by clicking the Summary tab of the user or group's profile.

# How to Assign Tokens to Users

In a Sun Ray environment, users will take advantage of smart cards (tokens) to initiate a session on a Sun Ray thin client (DTU). With VDI 3.1, you can assign a token to a user. It is also possible to assign desktops directly to specific tokens. Once tokens have been created, they can be assigned to pools and desktops.

## VDI Manager Steps

1. Sign into the VDI Manager.
    a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
    b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Users tab and Users and Groups entry in the left sidebar.

3. Search for a known user in the user directory.

4. Click on the user's name, and then select the Token tab in their profile.

5. Assign the token.
    - If you are assigning a new token, click New in the Tokens table. Then Enter the ID of the new token (e.g. Payflex.500d9b8900130200).
    - If you are assigning an existing token, select Add in the Tokens table. Then search for the desired token.

> ℹ️ Token IDs can be copied directly from the SRSS Admin GUI (see the Tokens tab and display Currently Used Tokens).

## CLI Steps

1. Open a terminal window and sign into the server with root credentials.
   For a multi-host configuration, use one of the VDI Secondary hosts.

2. Assign a token.
    - Assign a new token to a user.

      ```
      # /opt/SUNWvda/sbin/vda token-create -p token-id=<token ID>,user=<user ID>
      ```

        - Example – Creating a new token and assigning it to a user

          ```
          # /opt/SUNWvda/sbin/vda token-create -p
          token-id=Payflex.600a7c5600130200,user=jd123456
          Token Payflex.600a7c5600130200 created
          ```

    - Assign an existing token to a user.

      ```
      # /opt/SUNWvda/sbin/vda token-create -p token-id=<token ID>,user=<user ID>
      ```

        - Example – Assigning an existing token to a user

```
# /opt/SUNWvda/sbin/vda token-setprops -p user=jd123456
Payflex.600a7c5600130200
Token properties updated
```

# How to Assign Tokens to Desktops or Pools

You can assign tokens to desktops or desktop pools. This is similar to assigning desktops to users, however, a single user can potentially own multiple tokens (smart cards). By assigning tokens to desktops, users are able to easily switch between the assigned desktops by just inserting different smart cards into the DTU.

## VDI Manager Steps

1. Sign into the VDI Manager.
   a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.
   b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Users tab and Tokens entry in the left sidebar.

3. Select a token from the Tokens table.

4. Assign the token.
   - If you are assigning a token to a desktop, click Add on the Assigned Desktops table. Then enter the ID of the token (e.g. Payflex.500d9b8900130200).
   - If you are assigning a token to a pool, click Add on the Assigned Pool table. Then enter the ID of the token (e.g. Payflex.500d9b8900130200).

> ℹ️ Token IDs can be copied directly from the SRSS Admin GUI (see the Tokens tab and display Currently Used Tokens).

Assigning desktops or pools to each token individually can be cumbersome. Thus Sun VDI also provides some predefined special tokens ("AnySmartCard.000" and "AnySunRayClient.000") which can be used to make default pool assignments. For example, if you assign a pool to the AnySmartCard.000 token, any user taking advantage of a smart card (regardless of the smart card ID) will get a desktop from this pool. Similar the AnySunRayClient.000 token allows to assign all Sun Ray clients (Sun Ray DTUs and Sun Desktop Access Clients) together to a pool. Any user will get a desktop from this pool if the Sun Ray client is used without a smart card.

## CLI Steps

1. Open a terminal window and sign into the server with root credentials.
   For a multi-host configuration, use one of the VDI Secondary hosts.

   - Assign a token to a desktop.
     a. List available desktops.

     ```
     # /opt/SUNWvda/sbin/vda pool-desktops <pool name>
     ```

     b. Assign the token to one of the listed desktops.

     ```
     # /opt/SUNWvda/sbin/vda token-assign --desktop=<desktop ID> <token ID>
     ```

   - Example – Assigning an existing token to a desktop

```
# /opt/SUNWvda/sbin/vda pool-desktops "Sales - EMEA"
NAME                            ID  MACHINE_STATE       STATUS
DN
OpenSolaris 2008.11 De 2
                                 2     Powered Off     Available
-

# /opt/SUNWvda/sbin/vda token-assign --desktop=2 Payflex.500d9b8900130200
Token Payflex.500d9b8900130200 assigned to desktop 2
```

or

- Assign a token to a pool.

```
# /opt/SUNWvda/sbin/vda token-assign --pool=<pool name> <token ID>
```

- Example – Assigning an existing token to a pool

```
# /opt/SUNWvda/sbin/vda token-assign --pool="Sales - EMEA" Payflex.500d9b8900130200
```

- Example – Assigning all smart cards to a pool

```
# /opt/SUNWvda/sbin/vda token-assign --pool="Sales - EMEA" AnySmartCard.000
```

## How to Create Bulks of Tokens

It is possible to create a number of tokens at once using the token-create subcommand.

The token-create subcommand can take an input file containing the tokens to create and the user associated with the token if needed.

### Usage

```
Options:
-f <token-file>, --file=<token-file>
A CSV file containing the properties of the tokens to
be created. Format of the file is: <token-id> <comment>
<userid>
-w, --write Overwrite existing tokens, option to be used with the
token-file option
```

The format of the token file is CSV with the following values:

- token-id: the id of the smart card, this value is mandatory.
- comment: a comment about the token that can be used as a user friendly description of the token. This value maybe empty.
- userid: the user id of a user from the user directory, to be associated with the token. This value maybe empty.

### Example

The following example shows a valid csv file for token creation and uses the file to create the tokens and their association to users.

```
example% cat /tokens.csv
mo12.345,"token for Mary O'Leary",moleary
js46.23,"token for user John Smith",jsmith
x34.45,"token without any associated user",
example% /opt/SUNWvda/sbin/vda token-create -f /tokens.csv
example% /opt/SUNWvda/sbin/vda token-search
NAME USER DN
mo12.345 Mary O'Leary cn=Mary O'Leary,ou=people
js46.23 John Smith cn=John Smith,ou=people
x34.45 - -
```

# How to Create Automated Administration Scripts

The `/opt/SUNWvda/sbin/vda` CLI can be used in scripts for automated administration.

## Reading the Return Code

The `/opt/SUNWvda/sbin/vda` returns the following exit codes:

- 0: Successful completion
- 1: An error occurred
- 2: Invalid command line options or arguments were specified

## Waiting for a Job to Finish

Some `vda` subcommands return immediately but start an action in the background, a job.
The subcommand `job-wait` allows to synchronously wait for a specific job to be completed.

```
# /opt/SUNWvda/sbin/vda job-wait --help
Wait until the job ends

Usage:
 vda job-wait [-t <timeout> | --timeout=<timeout>] <job>
 -?, --help               Print this help list
Options:
 -t <timeout>, --timeout=<timeout>
                          Timeout in seconds to wait
Operand:
*<job>                    The id of the job
'*' denotes mandatory parameters.
```

## Parsing the Output of the CLI

A number of subcommands support a parseable option so that the output is formatted for easy parsing: as a list of lines of colon-separated (':') fields.

The syntax of the option is:

```
 -x, --parseable          Display output suitable for programmatic parsing.
```

### user-search

Search for users/groups in the user directory that match the specified search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Name of the user/group | string |
| Kind of object | User / Group |
| DN of the user/group | string |

### user-show

Show the desktops available for the user.

Parseable Output in the case of a user: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | User / Token <token> / Group <group_name> / Custom Group <group_name> |

Parseable Output in the case of a group: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |

### user-desktops

Show the desktops assigned to the user.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | flexible / personal |
| Is Default Desktop | true / false |

### group-list

Lists all custom groups.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Custom Group Name | string |

### group-show

Show the pools assigned to the custom group.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|

| Pool Name | string |
|---|---|

### token-search

Search for tokens that match the search criteria.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Token | string |
| Name of the Associated User | string |
| DN of the Associated User | string |

### token-show

Show the desktops available for the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Desktop Name | string |
| Desktop ID | integer |
| Kind of Assignment | `User` / `Token` / `Group <group_name>` / `Custom Group <group_name>` |

### token-desktops

Show the desktops assigned to the token.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | integer |
| Pool Name | string |
| Type of Assignment | `flexible` / `personal` |
| Is Default Desktop | `true` / `false` |

### pool-list

List all pools.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Pool Name | string |
| Type of Desktop Assignment | `Personal` / `Flexible` |
| Number of Desktops | integer |
| Desktop Provider Name | string |

## pool-show

Show detailed information about the pool.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
| --- | --- |
| Assignment Status | `Enabled` / `Disabled` |
| Type of Desktop Assignment | `Personal` / `Flexible` |
| Desktop Provider Name | string |
| Cloning Status | `Enabled` / `Disabled` |
| Template | `None` / string |
| Number of Cloning Jobs | integer |
| Number of Available Desktops | integer |
| Number of Assigned Desktops | integer |
| Total Number of Desktops | integer |

## pool-desktops

List all desktops from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
| --- | --- |
| Desktop Name | string |
| Desktop ID | long |
| Machine State | `Running` / `Powered Off` / `Suspended` / `Unknown` |
| Desktop State | `Available` / `Used` / `Idle` / `Unresponsive` / `Reserved` / etc. |
| DN of Assigned User | string |

## pool-templates

List all templates from the pool.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
| --- | --- |
| Template Name | string |
| Template ID | long |
| Machine State | `Running` / `Powered Off` / `Suspended` / `Aborted` / `Unknown` |
| Master Revision | string |
| Cloned Desktops | string |

## template-revisions

List the revisions of the template.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Revision Name | string |
| Revision ID | long |
| Creation Date | timestamp |
| Is It Master | yes / no |
| Cloned Desktops | string |

## provider-list

List all desktop providers.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Provider Name | string |
| Provider Type | Sun VirtualBox / VMware vCenter / Microsoft Hyper-V / Microsoft Remote Desktop |
| Total Number of Desktops | integer |
| Number of Used Desktops | integer |
| CPU Usage | xx% (x.x GHz/MHz) |
| Memory Usage | xx% (x.x GB/MB) |
| Storage Usage | xx% (x.x GB/MB) |

## provider-list-hosts

List all hosts for the VirtualBox desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Host Name | string |
| Status | OK / Unresponsive / etc. |
| Enabled | Enabled / Disabled |
| CPU Usage | xx% (x.x GHz/MHz) |
| Memory Usage | xx% (x.x GB/MB) |
| Number of Desktops | integer |

## provider-list-storage

List all storage servers for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Status | OK / Unresponsive / etc. |
| Enabled | Enabled / Disabled |

| | |
|---|---|
| ZFS Pool | string |
| Capacity | `xxx.x GB` |
| Usage | `xx.x GB` |
| Number of Desktops | integer |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Storage Name | string |
| Storage ID | string |
| ZFS Pool | string |
| Capacity | `xxx.x GB` |
| Usage | `xx.x GB` |
| Number of Desktops | integer |

### provider-list-templates

List the templates for the desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Template Name | string |
| Template ID | string |
| Path | string |

### provider-list-unmanaged

List the desktops from the virtualization platform that are not managed by any desktop provider.

Parseable Output for VirtualBox and Hyper-V providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Host Name | string |
| Desktop Name | string |
| Desktop ID | long |

Parseable Output for VMware vCenter providers: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Desktop Name | string |
| Desktop ID | string |

## provider-list-networks

List all networks for the desktop provider.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Subnet Label | String |
| Subnet Address | String |
| Availability | `All Hosts` / `Not on: <comma_separated_list_of_hosts>` |

## job-list

List the existing jobs.

Parseable Output: list of lines with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| ID of the Job | integer |
| Cancellable | 'C' if the job can be cancelled |

## job-show

Show the job details.

Parseable Output: one line with the following values separated by a colon (':').

| Value | Data Format |
|---|---|
| Job Title | `Cloning Desktop <desktop_name>` / `Recycling Desktop <desktop_name>` / `Starting Desktop <desktop_name>` / `Powering Off Desktop <desktop_name>` / `Shutting Down Desktop <desktop_name>` / `Restarting Desktop <desktop_name>` / `Deleting Pool <pool_name>` / etc. |
| Target of the Job | string |
| Status of the Job | `Queued` / `Running` / `Completed` / `Failed` / `Cancelling` / `Cancelled` |
| Start Time | hh:mm:ss |
| End Time | hh:mm:ss |
| Job Details | string |
| Cancellable | `true` / `false` |

Contents

# Sun Ray Access (All Topics)

## About Sun Ray Software in VDI 3.1

Since version 3.0, VDI has included a common installer for the VDI Core Software and the Sun Ray Software components which installs and configures a default version of Sun Ray Software for VDI. The common installer is a significant improvement for installation and configuration of a full VDI solution, especially for administrators who are less familiar with Sun Ray Software. Administrators who have a strong Sun Ray Software background may choose to change the defaults of the configuration that is installed for VDI by reviewing the VDI Defaults page and using the included links to access relevant Sun Ray Software information.

The Sun Ray Software for VDI 3.1 is based on Sun Ray Software 5, which includes several new features that can also be useful in VDI. The following sections provide an overview of these features and their default configurations with the common installer.

### USB Device Redirection

USB redirection is a Sun Ray Software 5 feature that enables users to access USB devices connected to a Sun Ray DTU from their Windows XP sessions, provided that the appropriate device drivers are installed on the Windows server. USB redirection is automatically enabled during VDI Core configuration and is supported as part of the Per Pool RDP Settings. It can be enabled or disabled using the VDI Manager or CLI. See the Feature Overview table on the How to Configure RDP Options Per Pool page for more information.

For the most up-to-date details about USB Redirection, refer to the About USB Device Redirection page in the SRWC 2.2 Information Center.

### Sun Desktop Access Client

The Sun Desktop Access Client is a software application that runs on common client operating systems and provides the ability to connect to a desktop session running on a Sun Ray server. Users can switch between their Sun Ray DTU and any supported Desktop Access Client enabled PC without using smart cards. In other words, a user can install and run the Sun Desktop Access Client instead of relying only on a Sun Ray Desktop Unit (DTU) for session access. For example, a user could connect to the same Sun Ray session from a PC laptop or desktop at home and a Sun Ray DTU at the office.

For the most up-to-date details about Sun Desktop Access Clients, refer to the About Sun Desktop Access Clients page in the SRSS 4.2 Information Center.

## About the Bundled Sun Ray Kiosk Session

Sun Ray Software is typically used to serve standard UNIX desktop sessions. However, other session types can be easily supported taking advantage of the Sun Ray Kiosk mode. Sun VDI 3.1 comes with a predefined Kiosk session (called Sun Virtual Desktop Access - VDA) that uses the Sun Ray Windows Connector to establish a remote desktop protocol (RDP) connection to a virtual machine.

A Sun Ray Kiosk session is initiated when a user inserts a smart card into a Sun Ray DTU. The new session will first bring up a login dialog asking for user name and password (and optionally a Windows domain). This type of authentication can be disabled if required (see Disabling Client Authentication).

> ℹ️ Sun VDI 3.1 does not require the usage of smart cards. Per default the Kiosk session is enabled for smart card and non-smart card access.

After successful authentication, the system will contact the Sun VDI service to determine the desktops assigned to the logged in user. If multiple desktops are available, the user will get a desktop selection dialog. Once the user has selected a desktop, the Sun Ray Windows Connector will startup and connect to the virtual machine running the desktop (see DTU Access).

## Sun Ray Kiosk Session Configuration

The appearance and behavior of the Kiosk session can be configured via a number of session parameters. These parameters can be split up into two groups: Settings specific for the VDA session (affecting the desktop selector dialog), and settings specific for the Sun Ray Windows Connector (aka uttsc) (affecting the quality of the RDP connection). The general syntax is:

```
<specific settings for desktop selector> -- <uttsc specific settings>
```

## Supported VDI Desktop Selector Parameters

Per default the login/desktop selector dialog will take advantage of the Java Runtime Environment bundled with Sun VDI. However, an alternative path can be specified using the "-j" option. We recommend to use Java 6 for the dialog to get better locale support and to take advantage of the latest improvements in the Java Swing area.

Other Kiosk parameters set default values for the input fields or hide/display certain UI elements in the dialog.

```
-n (--no-desktop-selector)    - Disables the desktop selector completely.
-d (--default-domain)         - Allows to preset domain input field.
-l (--list-of-domains)        - Preset the domain selector pulldown, e.g. -l
vdatest.germany,qa.ireland
-t (--timeout)                - Specifies the timeout applied after login (seconds)
-j (--java-home)              - Path to JRE used by the selector dialog.
-a (--allow-username-editing) - Allow users to login with a different user name (normally the user
name
                                field is readonly).
-h (--no-username-field)      - Always hide the user name input field.
-o (--no-domain-field)        - Always hide domain input field.
-w (--show-password-field)    - Always show password field
```

> ℹ️ Disabling the desktop selector
> If you disable the desktop login/selector dialog with the "-n" option, then users have no possibility to enter their password prior to accessing a desktop. Thus if you disable this dialog, you must also disable the client authentication at the same time - see Disabling Client Authentication.

## Supported Sun Ray Windows Connector (uttsc) Parameters

See the uttsc man page for a complete listing of the supported parameters. The list below is just an extract of the settings to illustrate the configuration options.

```
    -r sound:[low|high|off]     - Disable sound redirection from the server to the client or change the
    quality of

                                  transmitted sound. The sound quality in terms of bits per second can
    be specified.

                                  A "low" quality transmits 8khz and a "high" quality does 22.2 khz.
    By default, High

                                  quality sound is enabled.

    -A color depth              - Sets the colour depth for the connection (8, 15, 16  or 24).
                                  The  colour  depth  may be limited by the server configuration in
    which case

                                  the server configuration is honored.

    -E window-attribute         - Enable window attributes from the defined set. The available set of
    options which

                                  can be enabled are:
                                     wallpaper,  fullwindowdrag,  menuanimations,   theming,
                                     cursorshadow,  cursorsettings.
                                  Keeping these attributes disabled improves display performance
    especially over lower

                                  bandwidth  networks.  Multiple -E options can be specified for more
    than one attribute

                                  if required.
```

# How to Adapt the Bundled Sun Ray Kiosk Session

Use the instructions provided by the Sun Ray documentation to adapt the default Sun Ray Kiosk settings.
Your entry in the arguments field could look like the following:

```
    -d vdatest -j /usr/java6 -- -E wallpaper -E theming
```

For more about the default Sun Ray Kiosk settings, see the VDI Defaults page.

## Activating New Settings

The new settings will become active for every newly created Kiosk session. If you want to enforce the settings for existing sessions also, then you can perform a cold Sun Ray services restart. This will terminate all existing sessions and will create new Kiosk sessions as necessary.

Manager Steps

1. Switch to the Servers tab.
2. Select all servers in your Sun VDI environment.
3. Click on Cold Restart to initiate the Sun Ray services restart.
4. This operation can take up to several minutes.

CLI Steps
Refer to the Sun Ray Documentation for more information.

# How to Access Desktops Using a Sun Ray Client (with a VDI Desktop Selector)

The following screenshots illustrate how to access a desktop from an end-user perspective using Sun Ray clients (DTUs or SRDC instances).

Starting with Sun VDI 3.1 all users must authenticate themselves before getting access to any desktops. Also new is the possibility to select between multiple desktops. This behavior can be configured using the information on the How to Adapt the Bundled Sun Ray Kiosk Session page.

## Steps

1. Log into Sun VDI.
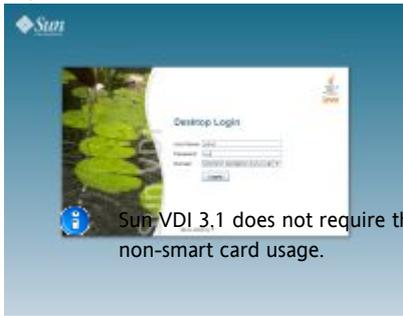
Insert a smart card (token) that has been assigned to a pool, or a desktop directly (as described before) into a Sun Ray DTU that is connected to a Sun VDI host. It should display a login screen, after a short while. You must provide your user name and password (and optionally a Windows domain).

Sun VDI 3.1 does not require the usage of smart cards. Per default desktop access is enabled for smart card and non-smart card usage.

2. Select a desktop or pool.
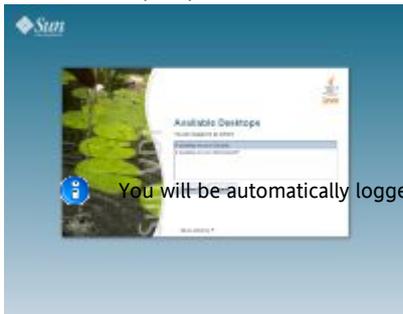
After successful authentication, the system will determine the desktops (and pools) assigned to you. If multiple desktops are available, you will get a desktop selection dialog. The selection screen will be skipped, if there is only one desktop available to you.

You will be automatically logged out, if you do not select a desktop within three minutes (the default timeout).

3. Work with the desktop.

Once you have selected a desktop, the Sun Ray Connector for Windows OS will startup and will display your desktop.
At any time you can disconnect from your desktop by moving your mouse up to the top of the screen. A remote desktop pulldown menu will appear. Hitting the "X" from the menu will disconnect you from the current desktop session and the desktop selection menu (or the login screen) will appear again.

There is also a Disconnect button available in the Windows start menu, for desktops connected via Windows RDP. Desktops connected via VirtualBox RDP (VRDP) will not offer this button.

## How to Access Desktops Using a Sun Ray Client (without a VDI Desktop Selector)

The VDI desktop selector dialog can be disabled with the "-n" Kiosk session option. In this setup users are always connected to their default desktop without the need to pass any other Sun VDI dialogs.

If you disable the desktop selector dialog, users do not have the possibility to enter their password prior to accessing a desktop. Thus to make this setup work, you must also disable the Sun VDI client authentication - see Disabling Client Authentication.

Steps

1. Start the desktop.

Insert a smart card (token) that has been assigned to a pool, or a desktop directly (as described before) into a Sun Ray DTU that is connected to a Sun VDI host. Sun VDI will determine the assigned default desktop and will start it up if necessary. During that time a wait screen is displayed.

ⓘ Sun VDI 3.1 does not require the usage of smart cards. Per default desktop access is enabled for smart card and non-smart card usage.

2. Log into the desktop.



It is good practice, if desktops are configured to always present their own login screen before displaying the actual desktop content. This way authentication is still required, but it is now performed on the guest OS level.
In this example you will get the standard Windows login screen. Depending on your guest OS configuration you must enter user name/password (and potentially the Windows domain).

3. Work with the desktop.



Once you have successfully logged in you will get your desktop content displayed. The behavior is the same as for a standard Windows PC.

## How to Access the Sun Ray Administration GUI

The Sun Ray Administration GUI is configured and accessible on each Sun VDI host. This allows easy modification of Sun Ray configuration settings such as Kiosk session parameters (see following section).

Steps

1. Go to `http://<server name>:1660`.
2. You will be re-directed to https and the web browser will ask you to accept the security certificate. After confirmation, you should get the login screen.
3. You must login as super user ("root") with corresponding password.

ⓘ Sun VDI 3.1 does not use the default "admin" user account that is normally configured as part of the Sun Ray Software installation.

# How to Change User Password

The desktop login/selector dialog allows end-users working from Sun Ray thin clients (DTUs) to update their password in the user directory.

> ℹ️ Password Change is not offered when Client Authentication is disabled.

## Supported User Directories

Sun VDI supports password change on the following directory servers:

- Active Directory (from Windows Server 2003 and 2008)
- Sun Directory Server

The authentication type selected to integrate the user directory with Sun VDI affects the password change functionality:

- Kerberos authentication and Public Key authentication allow end-users to change their password before it has expired as well as after expiration.
- LDAP Anonymous Authentication, LDAP Simple Authentication and LDAP Secure Authentication allow end-users to change their password before it has expired only. If the user password expires in such a configuration, the end-user will be required to update her password using a customer-provided process external to Sun VDI.

> ℹ️ A default restriction in Active Directory prevents password update from an LDAP Simple Authentication.

## The user password has already expired

If integrating with an Active Directory server using Kerberos authentication or Public Key authentication:

1. the end-user enters her login credentials in the login dialog
2. the system detects that the user password has expired and direct the user to the password change dialog where the user is offered to type her old and new passwords (new password needs to be entered twice)
3. after a successful password update, the user is authenticated with the new password and the system will offer the same screen as after a regular successful authentication.

If using an LDAP type of authentication:

1. the end-user enters her login credentials in the login dialog
2. the system detects that the user password has expired and displays an error message to the end-user
3. the end-user must use an alternate customer-provided process to update her password before to be able to log in again.

## The user password has not expired yet

> ℹ️ This functionality may only be accessed from the desktop selector dialog, which is not displayed to the end-user when only one desktop is applicable to her.

This functionality is offered with all types of authentication for the user directory (provided the directory server supports end-users to change their password):

1. the desktop selector dialog offers a More Options... menu at the bottom which contains a Change Password... entry
2. when clicking on Change Password..., the user gets directed to the password change dialog where she is offered to type her old and new passwords (new password needs to be entered twice)
3. the user may cancel her password change, she then goes back to the desktop selector screen without any change to take place
4. when the user confirms the password change, her password gets updated in the directory server and she then goes back to the desktop selector screen with a confirmation message.

## Troubleshooting

The update of the password may fail for the following reasons:

- the end-user does not type the right old password
- the new password does not comply to the password policy from the directory server (not allowed to reuse old password, password complexity not met...)
- if using Active Directory server, the Kerberos configuration does not allow password change. See How to Set Up Kerberos Authentication for help on setting up Kerberos authentication.
- the authentication type does not allow password change. See restrictions described in Supported User Directories

In case of trouble, increase the log level in order to get more information about the error in the `/var/cacao/instances/default/logs/cacao.0` log file.

# How to Disable Client Authentication

Starting with Sun VDI 3.1, all users must authenticate themselves before getting access to any desktop. Typically users will be asked for a user name/password combination (and optionally a Windows domain). The VDI service will then contact the user directory for the verification of the provided user credentials. If authentication succeeds the connection to the desired desktop will be established - otherwise it will be denied. The user name/password will also be forwarded to the guest OS running the desktop - this way users get automatically logged into their desktop without the need to potentially pass another login screen.

> ℹ️ For Sun VDI 3.1 automatic login will work for Windows RDP only - forwarding of user credentials does not work yet for VRDP and non-Windows OS.

Authentication on the VDI service level can be disabled if desired. However, special care needs then to be taken on the users' desktops setup to not open unwanted security holes. For example, it is good practice, if desktops are configured to always present their own login screen before displaying the actual desktop content. This way authentication is still required, but it is now performed on the guest OS level only. This setup also allows to take advantage of more advanced authentication techniques that are not supported out of the box by the VDI service.

> ℹ️ For security reasons it is recommended to leave authentication always enabled, unless the simple user name/password authentication does not satisfy your requirements.

## Steps

You can use the VDA administration CLI to configure, if authentication should be performed by the VDI service.

To check the currently configured authentication policy:

```
# /opt/SUNWvda/sbin/vda settings-getprops -p clientauthentication
```

To enable authentication (the default):

```
# /opt/SUNWvda/sbin/vda settings-setprops -p clientauthentication=Enabled
```

To disable authentication:

```
# /opt/SUNWvda/sbin/vda settings-setprops -p clientauthentication=Disabled
```

# How to Configure RDP Options Per Pool

With VDI 3.1 you can configure the RDP options to be used by Sun Ray sessions when users connect to their desktops.

## VDI Manager Steps

1. Sign into the VDI Manager.

a. Go to http://<server name>:1800 (or http://localhost:1800 if remote administration has been disabled), and use root user credentials. For a multi-host configuration, use one of the VDI Secondary hosts.

b. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

2. Select the Pools category and select the pool of interest.
   a. In the pool overview, select the Settings tab.
   b. In the Sun Ray section, click the Edit Sun Ray RDP Settings link.
   c. Enable the desired RDP settings and click Save.
   d. Click Back, and select the Use Customized Settings option in the Sun Ray section.
   e. Click Save.

## Available RDP Options

Sun Ray Connector for Windows OS (`uttsc`) supports a wide range of options allowing you to configure RDP connections from Sun Ray to your users' desktops.

VDI 3.1 allows you configure a subset of these options on a per pool basis. The following table lists the supported options. For details about how VDI's Sun Ray settings compare to the SRWC `uttsc` settings, refer to the VDI Defaults page.

| Name | Description | Default Value |
|------|-------------|---------------|
| General | | |
| Locale | Use this setting to identify the locale used for users' desktop sessions.<br>Any valid locale identifer may be specified, for example, `en-US` or `de-DE`. | en-US |
| Keyboard Layout | Use this setting to identify the keyboard type used for users' desktop sessions<br>Valid values for this setting include All Sun and PC USB Keyboards, Sun Type6 Japanese Keyboard, and Sun Korean Keyboard. | All Sun and PC USB Keyboards |
| Optimized Hotdesking | Use this setting to enable or disable optimized hotdesking behaviour. If enabled, Sun Ray sessions can be hotdesked without restarting `uttsc`. | Disabled |
| Windows Pulldown Header | Use this setting to enable or disable the Windows pulldown header. | Enabled |
| RDP Packet Data Compression | Use this setting to enable or disable the compression of RDP packet data. | Enabled |
| Appearance | | |
| Colour Depth | Use this setting to specify the preferred colour depth for users' desktop sessions.<br>Valid values for this setting are 8, 15, 16, 24 and 32.<br>Note: Colour depth may be limited by configuration of the desktop to which a user connects. In such cases the available colour depths of the desktop will take priority over the colour depth configured for the pool containing the desktop. | 32 |
| Theming | Use this setting to enable or disable theming for users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Desktop Background | Use this setting to enable or disable the desktop background for users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Show Window Contents While Dragging | Use this setting to enable or disable the ability to show complete window contents while dragging windows in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Transition Effects for Menus | Use this setting to enable or disable visual effects during the use of menus in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Pointer Shadow | Use this setting to enable or disable the use of pointer shadow in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |

| | | |
|---|---|---|
| Pointer Scheme | Use this setting to enable or disable the use of pointer schemes in users' desktop sessions.<br>Note: Disabling this setting can improve display performance. | Disabled |
| Sound | Use this setting to control sound quality in users' desktop sessions.<br>Valid values for this setting are "High" (to enable high quality sound), "Low" (to enable low quality sound) and "Off" (to disable sound). | High |
| Redirection | | |
| Smart Cards | Use this setting to enable or disable smart card redirection from a DTU to users' desktop sessions. | Disabled |
| USB | Use this setting to enable or disable USB redirection from a DTU to users' desktop sessions. | Enabled |
| Serial Devices | Use this setting to identify serial devices which should be redirected to users' desktop sessions.<br>Valid values for this setting are specified using the format <comport>=<device> where <device> identifies the serial device to be redirected and <comport> identifies the port (on the users' desktops) that <device> should be redirected to. | No serial devices are redirected by default. |
| Paths | Use this setting to identify paths (available on a VDI host) which should be redirected to drives on users' desktop sessions.<br>Valid values for this setting are specified using the format <drive name>=<path> where <path> identifies the path to be redirected and <drive name> identifies the drive (on the users' desktops) that <path> should be redirected to. | No paths are redirected by default. |
| Printers | Use this setting to identify printer queues which should be redirected to users' desktop sessions.<br>Valid values for this setting are specified using the format <printer>=[<driver>] where <printer> identifies the printer queue to be redirected and <driver> identifies a printer driver to be used for the printer on users' desktop sessions. If <driver> is omitted, a simple PostScript driver is used by default. | No printer queues are redirected by default. |

# VDI Defaults

The VDI Core aims to simplify the management of multiple (typically independent) software components. You should be able to set up a functional and customized VDI setup within the VDI options with the documentation provided. But, there may also be times when you want to know the VDI defaults on the bundled software. The information on this page is intended to be used as a reference.

## Sun Ray Windows Connector (SRWC) 2.2

### VDI Core Configuration

| Command | Details | Main Page |
|---|---|---|
| `/opt/SUNWuttsc/sbin/uttscadm -c` | Enables SRWC if the Kiosk and LAN settings succeed. | About VDI Core Configuration |

### Sun Ray RDP Settings (uttsc)

| VDI Name | uttsc option equivalent | Main Page |
|---|---|---|
| Locale | -l <locale> | How to Configure RDP Options Per Pool<br>Solaris Sun Ray User Commands - man(1) |
| Keyboard Layout | -k <keyboard> | How to Configure RDP Options Per Pool<br>Solaris Sun Ray User Commands - man(1) |
| Optimized Hotdesking | -O | How to Configure RDP Options Per Pool<br>Solaris Sun Ray User Commands - man(1) |
| Windows Pulldown Header | -b | How to Configure RDP Options Per Pool<br>Solaris Sun Ray User Commands - man(1) |

| RDP Packet Data Compression | -z | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
|---|---|---|
| Color Depth | -A <color depth> | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Theming | -E theming | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Desktop Background | -E wallpaper | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Show Window Contents While Dragging | -E fullwindowdrag | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Transition Effects for Menus | -E menuanimations | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Pointer Shadow | -E cursorshadow | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Pointer Scheme | -E cursorsettings | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Sound | -r sound | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Smart Cards | -r scard:on | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| USB | -r usb:on | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Serial Devices | -r comport: | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Paths | -r disk: | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |
| Printers | -r printer: | How to Configure RDP Options Per Pool Solaris Sun Ray User Commands - man(1) |

## Sun Ray Server Software (SRSS) 4.2

### VDI Core Configuration

| Command | Details | Main Page |
|---|---|---|
| ```/opt/SUNWut/sbin/utconfig``` | Configures basic SRSS settings: <br> * Administrator password <br> * Server for a FOG <br> * FOG's signature | About VDI Core Configuration |
| ```/opt/SUNWut/lib/support_lib/srwa_config update``` | Configures SRSS Web Administration: <br> * Tomcat's home directory <br> * http ports (1660, 1661) <br> * Webservice's user name(utwww) <br> * Remote access (enabled) | About VDI Core Configuration |
| ```/opt/SUNWkio/bin/kioskuseradm create -l utku -g utkiosk -i auto -u -c``` | Configures Kiosk user accounts | About VDI Core Configuration |

| Command | Details | Main Page |
|---|---|---|
| ```<br>/opt/SUNWut/lib/utrcmd -n<br>/opt/SUNWut/sbin/utreplica -p -a<br>/opt/SUNWut/sbin/utreplica -s<br>``` | Replicates from Primary to Secondary hosts | About VDI Core Configuration |
| ```<br>/opt/SUNWut/sbin/utadm -L on<br>``` | Enables LAN access | About VDI Core Configuration |
| ```<br>/opt/SUNWut/sbin/utadminuser -a root<br>/opt/SUNWut/sbin/utadminuser -d admin<br>```<br><br>Additionally, the following line is commented out in the file `/etc/pam.conf`:<br><br>```<br>utadmingui auth sufficient<br>/opt/SUNWut/lib/pam_sunray_admingui.so.1<br>``` | Allows root user access | About VDI Core Configuration |
| ```<br>/opt/SUNWut/sbin/utkiosk -i session -f<br>``` | Sets Kiosk Session value to vda | About VDI Core Configuration |
| ```<br>/opt/SUNWut/sbin/utpolicy -a -g -z both -k both -m<br>``` | Sets Kiosk Policy for both card users and non-card users | About VDI Core Configuration |

## VDI Core

VDI Core Configuration

| Command | Details | Main Page |
|---|---|---|
| ```<br>svc://application/rdpbroker<br>``` | Creates RDP Broker SMF service | About VDI Core Configuration |
|  | Configures VDA webservice:<br>* Ports are configured (1800 / 1801)<br>* `webuser` is set to `noaccess`<br>* Remote access is enabled | About VDI Core Configuration |
| ```<br>cacaoadm stop -f<br>``` | Stops CACAO | About VDI Core Configuration |
| ```<br>cacaoadm set-param java-flags=-Xms4M -Xmx256M<br>-Dcom.sun.management.jmxremote -Dfile.encoding=utf-8<br>``` | Sets java and file-encoding flags | About VDI Core Configuration |

| | Starts CACAO | About VDI Core Configuration |
|---|---|---|
| ```cacaoadm start``` | | |
| ```cacaoadm enable -i default``` | Sets CACAO to start at boot | About VDI Core Configuration |

## System Preparation

| Command | Details | Main Page |
|---|---|---|
| ```sysprep.exe -mini -reseal -activated -quiet``` | On Windows XP virtual machines | How to Enable System Preparation for Windows Templates (VirtualBox and Hyper-V) |
| ```sysprep.exe -generalize -oobe -shutdown -quiet``` | On Windows Vista and Windows 7 virtual machines | How to Enable System Preparation for Windows Templates (VirtualBox and Hyper-V) |

Contents

# Secure Web Access (SGD) (All Topics)

## How to Set Up Sun Secure Global Desktop Software

Sun VDI can be also used with Sun Secure Global Desktop software (SGD) 4.41.

Steps

1. Install SGD 4.41 or SGD 4.5.
   For detailed instructions, see the Sun Secure Global Desktop Software 4.41 Collection.

   > The mechanism for accessing desktops via Sun Secure Global Desktop has changed since Sun VDI 2. The 'My Desktop' Application Object and corresponding expect script is no longer required for Sun VDI 3.1.

2. Replace the SGD Terminal Service Client. (SGD 4.41 Only)
   The default SGD Terminal Service Client (ttatsc) does not support RDP redirection as required by the Sun VDI Core. Thus, you need to replace this binary with an updated version included in the Sun VDI Core installation.
   Replace the default SGD Terminal Services Client provided by SGD (found at `/opt/tarantella/bin/bin/ttatsc`) with the version provided by the Sun VDI RDP Broker (found at `/opt/SUNWrdpb/supplemental/ttatsc/<Installation_Platform>/ttatsc`).

3. Create a Windows Application Object.

You need to create a Windows application object to offer users an easy way to access the desktops managed by Sun VDI. This can be done using the tarantella CLI or using the SGD Administration console.

For example, to create a full screen kiosk Windows application using the SGD command line enter the following command:

```
# /opt/tarantella/bin/tarantella object new_windowsapp --name
".../_ens/o=applications/cn=Sun VDI Desktop" \
--width 1200 --height 1000 --maxinstances 1 --login windows.exp --displayusing kiosk
--maximize true
```

4. Configure an application server for the new Application Object.
   If the SGD server and the VDI server are the same machine, no further commands are necessary. An application object will use the SGD server as the application host by default if an explicit application host isn't specified.
   a. However, if you need to create a host object, use the following command:

   ```
   /opt/tarantella/bin/tarantella object new_host --name
   ".../_ens/o=appservers/cn=hostname" --address "hostname.domain.com"
   ```

   For more information, refer to the SGD documentation.

   b. To assign the host object to an application object:

   ```
   /opt/tarantella/bin/tarantella object add_host --name ".../_ens/o=applications/cn=Sun
   VDI Desktop" --host ".../_ens/o=appservers/cn=hostname"
   ```

   For more information, refer to the SGD documentation.

5. Assign the new Application Object to the users that need to access a Sun VDI Desktop.
   By default, all the users are assigned to the Applications group so an easy way to do so is to add the newly created application object to the Applications group:

   ```
   # /opt/tarantella/bin/tarantella object add_member --name
   ".../_ens/o=applications/cn=Applications" \
   --member ".../_ens/o=applications/cn=Sun VDI Desktop"
   ```

   > ⓘ Creating the Application Object and adding the object to the Applications group can also be done via the SGD Administration Console. Please see the SGD documentation for more details on adding Application Objects to SGD.

## How to Access Desktops with SGD Web Access

1. Log into the SGD webtop.
   Use the URL `http://<ssgd_server>/`) as the user who has been assigned the pool/desktop. The Windows Application Object that you've previously created should appear in the list of applications on the left-hand side.
2. Click the Application Object.
   You should be asked for user credentials. Then a Windows session for the user will appear on the assigned desktop.

## About the Bundled RDP Broker

Sun VDI 3.1 includes a built-in RDP broker that allows easy desktop access leveraging the Remote Desktop Protocol (RDP). This way users can take advantage of existing RDP clients (for example, the remote desktop connection in Windows XP) for accessing desktops.

How Does it Work?

1. The RDP client first contacts the Sun VDI RDP broker (passing over any information like username, password, etc).
2. The RDP broker will then contact the VDI service on behalf of the client and will ask to startup the desired desktop.
3. The VDI service will first verify the username/password combination (if client authentication is enabled on the VDI service side - that is the default, see How to Disable Client Authentication).
4. If authentication succeeds, the corresponding desktop will be started up and the VDI service returns the IP and optionally RDP port of the virtual machine (VM) running the desktop.
5. This information is used by the RDP broker to construct an RDP Server Redirection Packet containing either:

- the VM host/IP address as the server to redirect to (if using Windows RDP, as done for VMware Infrastructure 3)
- or a routing token containing encoded IP address and RDP port information (if using the VirtualBox RDP (aka VRDP))

The latter is necessary, because VRDP does not use the standard Windows RDP port. Thus the RDP broker needs to hand back both the IP and the RDP port information. For details of the routing token encoding, see the 'Routing Token Format' section of 'Session Directory and Load Balancing Using Terminal Server'.

Finally this RDP redirection packet is sent back to the RDP client and the client will redirect accordingly.

## Supported RDP Clients

RDP clients that support all the above mentioned mechanism and that have been tested with Sun VDI are:

- the Microsoft terminal services client (aka remote desktop connection) as included in Windows XP and Windows Vista
- the Sun Ray Connector for Windows OS (aka uttsc)
- the SGD terminal services client (aka ttatsc, using the updated version delivered as part of Sun VDI)

Other clients may work, but have not been tested by QA.

## Security Considerations

Sun VDI 3.1 now authenticates users each time they sign into their desktop. If you would prefer to disable this feature, please see How to Disable Client Authentication.

# How to Disable Client Authentication

Starting with Sun VDI 3.1, all users must authenticate themselves before getting access to any desktop. Typically users will be asked for a user name/password combination (and optionally a Windows domain). The VDI service will then contact the user directory for the verification of the provided user credentials. If authentication succeeds the connection to the desired desktop will be established - otherwise it will be denied. The user name/password will also be forwarded to the guest OS running the desktop - this way users get automatically logged into their desktop without the need to potentially pass another login screen.

> ℹ️ For Sun VDI 3.1 automatic login will work for Windows RDP only - forwarding of user credentials does not work yet for VRDP and non-Windows OS.

Authentication on the VDI service level can be disabled if desired. However, special care needs then to be taken on the users' desktops setup to not open unwanted security holes. For example, it is good practice, if desktops are configured to always present their own login screen before displaying the actual desktop content. This way authentication is still required, but it is now performed on the guest OS level only. This setup also allows to take advantage of more advanced authentication techniques that are not supported out of the box by the VDI service.

> ℹ️ For security reasons it is recommended to leave authentication always enabled, unless the simple user name/password authentication does not satisfy your requirements.

## Steps

You can use the VDA administration CLI to configure, if authentication should be performed by the VDI service.

To check the currently configured authentication policy:

```
# /opt/SUNWvda/sbin/vda settings-getprops -p clientauthentication
```

To enable authentication (the default):

```
# /opt/SUNWvda/sbin/vda settings-setprops -p clientauthentication=Enabled
```

To disable authentication:

```
# /opt/SUNWvda/sbin/vda settings-setprops -p clientauthentication=Disabled
```

Contents

- How to Access Desktops with Microsoft RDC
- About the Bundled RDP Broker
- How to Disable Client Authentication

# Remote Desktop Client Access (RDC) (All Topics)

## How to Access Desktops with Microsoft RDC

Sun VDI 3.1 includes a built-in RDP broker that allows easy desktop access leveraging the Remote Desktop Protocol (RDP). This way users can take advantage of existing Windows PCs for accessing desktops. There is typically no need to install any additional software on your PC. Both Windows XP and Windows Vista provide out of the box the necessary functionality. The following screenshots illustrate how to access a desktop from an end-user perspective using Windows XP.

Steps

1. Open a remote desktop connection.
   a. Click on Start -> All Programs -> Accessories -> Remote Desktop Connection.
   b. In the dialog, specify as Computer the name, or IP address of the host running Sun VDI 3.1.
   c. Specify the user name, and optionally Windows domain. Click Connect.
   d. A popup dialog will come up asking for the user password. Enter the password and click OK.
   e. After a while, the desktop should be displayed, and be ready to use.



> ⓘ  The remote desktop connection on your computer might be configured for performance optimization. Thus certain elements like desktop background, theming, menu and window animations might not be displayed in your setup. You can easily adapt these settings (see Experience tab of the remote desktop connection) to meet your personal requirements.

2. Access a specific desktop or pool.
   If multiple desktops are assigned to a user, then Sun VDI will connect to the default desktop (which can be defined using the Sun VDI

Manager).

Alternatively, it is possible to specify the desired desktop or pool when opening the remote desktop connection. Just enter the user name, followed by the pool name, and the optional desktop ID using the following syntax:

```
<username>::pool=<poolname>[,desktop=<desktopId>]
```

It is usually sufficient to just specify the pool name. However, if you have multiple desktops assigned from the same pool, you must specify both the pool name and the desktop ID.

> ⓘ Desktop identifiers can be listed via the Sun VDI administration CLI executing /opt/SUNWvda/sbin/vda user-desktops <username>
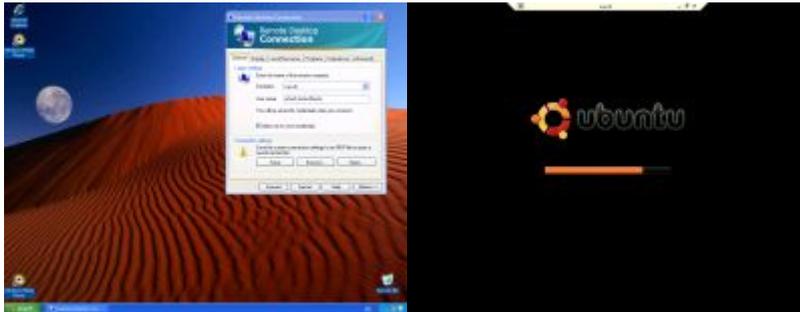
 If you frequently switch between various desktops, then it is convenient to store the remote desktop connection settings for each desktop in an RDP file (see Connection Settings -> Save As). You can then create shortcuts to these files allowing you to initiate a connection via a simple mouse double-click.

# About the Bundled RDP Broker

Sun VDI 3.1 includes a built-in RDP broker that allows easy desktop access leveraging the Remote Desktop Protocol (RDP). This way users can take advantage of existing RDP clients (for example, the remote desktop connection in Windows XP) for accessing desktops.

## How Does it Work?

1. The RDP client first contacts the Sun VDI RDP broker (passing over any information like username, password, etc).
2. The RDP broker will then contact the VDI service on behalf of the client and will ask to startup the desired desktop.
3. The VDI service will first verify the username/password combination (if client authentication is enabled on the VDI service side - that is the default, see How to Disable Client Authentication).
4. If authentication succeeds, the corresponding desktop will be started up and the VDI service returns the IP and optionally RDP port of the virtual machine (VM) running the desktop.
5. This information is used by the RDP broker to construct an RDP Server Redirection Packet containing either:

- the VM host/IP address as the server to redirect to (if using Windows RDP, as done for VMware Infrastructure 3)
- or a routing token containing encoded IP address and RDP port information (if using the VirtualBox RDP (aka VRDP))

The latter is necessary, because VRDP does not use the standard Windows RDP port. Thus the RDP broker needs to hand back both the IP and the RDP port information. For details of the routing token encoding, see the 'Routing Token Format' section of 'Session Directory and Load Balancing Using Terminal Server'.

Finally this RDP redirection packet is sent back to the RDP client and the client will redirect accordingly.

## Supported RDP Clients

RDP clients that support all the above mentioned mechanism and that have been tested with Sun VDI are:

- the Microsoft terminal services client (aka remote desktop connection) as included in Windows XP and Windows Vista
- the Sun Ray Connector for Windows OS (aka uttsc)
- the SGD terminal services client (aka ttatsc, using the updated version delivered as part of Sun VDI)

Other clients may work, but have not been tested by QA.

Security Considerations

Sun VDI 3.1 now authenticates users each time they sign into their desktop. If you would prefer to disable this feature, please see How to Disable Client Authentication.

# How to Disable Client Authentication

Starting with Sun VDI 3.1, all users must authenticate themselves before getting access to any desktop. Typically users will be asked for a user name/password combination (and optionally a Windows domain). The VDI service will then contact the user directory for the verification of the provided user credentials. If authentication succeeds the connection to the desired desktop will be established - otherwise it will be denied. The user name/password will also be forwarded to the guest OS running the desktop - this way users get automatically logged into their desktop without the need to potentially pass another login screen.

> For Sun VDI 3.1 automatic login will work for Windows RDP only - forwarding of user credentials does not work yet for VRDP and non-Windows OS.

Authentication on the VDI service level can be disabled if desired. However, special care needs then to be taken on the users' desktops setup to not open unwanted security holes. For example, it is good practice, if desktops are configured to always present their own login screen before displaying the actual desktop content. This way authentication is still required, but it is now performed on the guest OS level only. This setup also allows to take advantage of more advanced authentication techniques that are not supported out of the box by the VDI service.

> For security reasons it is recommended to leave authentication always enabled, unless the simple user name/password authentication does not satisfy your requirements.

Steps

You can use the VDA administration CLI to configure, if authentication should be performed by the VDI service.

To check the currently configured authentication policy:

```
# /opt/SUNWvda/sbin/vda settings-getprops -p clientauthentication
```

To enable authentication (the default):

```
# /opt/SUNWvda/sbin/vda settings-setprops -p clientauthentication=Enabled
```

To disable authentication:

```
# /opt/SUNWvda/sbin/vda settings-setprops -p clientauthentication=Disabled
```