Command Line Utilities Reference
ZENworks® 11 Support Pack 2

March 2013

Novell®

Contents

About This Guide

This *Novell ZENworks 11 Command Line Utilities Reference* includes information to help you use Novell ZENworks 11 utilities such as zman, zac, and zeninfocollect.

The information in this guide is organized as follows:

Audience

This guide is intended for ZENworks administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Additional Documentation

ZENworks 11 Configuration Management is supported by other documentation (in both PDF and HTML formats) that you can use to learn about and implement the product. For additional documentation, see the ZENworks 11 SP2 documentation Web site (http://www.novell.com/documentation/zenworks11)

1 ZENworks Command Line Utilities

The following sections give information on the Novell ZENworks 11 SP2 command line utilities:

zman(1)

Name

zman - The command line interface to manage Novell ZENworks 11 products, including , Asset Management, Configuration Management, Endpoint Security Management, and Full Disk Encryption.

Description

The zman command line interface provides quick access to most ZENworks management operations. By utilizing zman commands in scripts, you can automate many ZENworks management tasks.

The following sections provide information about all of the commands provided for the ZENworks 11 SP2 products. The commands apply to all products, with the following exceptions:

- **Asset Management commands:** Apply only to Asset Management.
- **Bundle commands:** Apply only to Configuration Management.
- **Imaging commands:** Apply only to Configuration Management.
- **Endpoint Security Management commands:** Apply only to Endpoint Security Management.
- **Full Disk Encryption commands:** Apply only to Full Disk Encryption.

Syntax

```
zman <command> [-options] [arguments]
```

In general, zman commands have both a short form and a long form. The long form is assembled in the form category-action. For example, the command registration-list lists all registrations in a registration folder. There are many commands, but most of them are easy to remember because they are grouped under categories (workstation, server, bundle, policy, administrator, registration, etc) and most actions (list, create, modify, delete) to perform on them are common.

The abbreviated form of each command uses one letter from each word in the long form. In this manner, admin-list is shortened to al. There are some exceptions to this rule to maintain the uniqueness of the short names. For example, bc is the short form of bundle-create, and bcp is the short form of bundle-copy.

A command takes options and arguments. Options can be specified anywhere in the command. Arguments are dependent on position.

Options

Options are of two types: switches and flagged options. Both have a short and long form. The short form is preceded by a single hyphen and the long form is preceded by two hypens. For example, -n and --namefilter are the short and long forms of a flagged option used to filter the display of results by name for the bundle-list command.

Switch: A parameter that does not take any value. Its presence alone conveys the meaning. For example, specifying the –recursive switch in bundle-list indicates that the contents of the subfolders should also be displayed.

Flagged Option: This parameter is used to set a value for an option. The value is given followed by a space or equals sign after the short or long flag of the option.

Example: `zman bundle-list -namefilter "bundle*" -r`

Arguments

Arguments for a command are usually ordered in the same way as the command itself. For example, for `server-add-bundle`, the server is named first and the bundle second.

Guide to Usage

This section contains a guide to general command formatting and conventions.

Administrators

Every action in zman is governed by the access limitations of the administrator. The initial Administrator account created during the initial installation has rights to all objects. Additional administrator accounts you create are granted read-only rights by default. These accounts must be explicitly granted rights to any objects they are to manage.

Help

The zman command line interface provides help at several levels. Pipeline the help through the operating system's native `more` command for scrollable output.

- At the highest level, lists all the categories of commands available in zman and the global options for the commands.

  Example: `zman`

- At the all category level, lists all the commands available in zman organized by categories (workstation, server, bundle, policy, etc.).

  Example: `zman -h`

- For each category, lists the commands available specific to that category.

  Example: `zman policy /?`

- At each command level, gives usage, detailed help, and examples for a particular command.

  Example: `zman policy-list-help`

- `-h`,`--help`,`?`,`-?`, and `/?` are the various help options available.

Folders

If no folder is specified for commands that take a folder argument, the command targets the root folder. To specify a folder path, list each folder from the root separated by a forward slash `(/)` . For example, if you have a folder named `folder1` in the root, containing a subfolder named `subfolder1`, you would reference this folder as `folder1/subfolder1`. Each specified folder must already exist.

Mandatory and Optional arguments

Arguments enclosed inside parentheses ( ) are mandatory and those enclosed by square brackets [ ] are optional.

Ellipsis (...)

An ellipsis indicates that a command accepts multiple entries of the last argument type. The arguments must be separated by space or commas, depending on the command. For example:

`zman server-add-bundle [options] (server) (bundle) [...]`

The ellipsis indicates that `server-add-bundle` can accept multiple bundles. In the command level help, the multiple entry argument help is given as `[(bundle) (bundle)... (bundle)]` for more clarity.

All commands that have multiple entries of the last argument also have the `-f|--folder` option. This option is convenient if all the entries of the argument are under the same folder. The path of the entries are calculated relative to this folder path. For example:

`zman server-add-bundle server1 bundlefolder/bundle1 bundlefolder/bundle2`

can also be given as:

`zman server-add-bundle server bundle1 bundle2 --folder bundlefolder`

Path of the ZENworks Objects

If the argument required is the path of a ZENworks object, the path must be specified relative to the root folder of that object separated by a forward slash (/), unless it is specified in the command help to enter the absolute path. For example:

`zman server-add-bundle serverfolder/server bundlefolder/bundle1`

where the absolute paths of the arguments are:

`/devices/servers/serverfolder/server` and `/bundles/bundlefolder/bundle1`

Exporting Objects to a File

You can use the zman commands such as `bundle-export-to-file`, `policy-export-to-file`, and `settings-export-to-file` to export bundles, policies, and settings definitions respectively to a specified file in the XML format. The file is created with the specified name and file extension. If the file extension is not specified, the `.xml` extension is added by default. You can use these files to create bundles, policies, and settings.

zman Service

You can now run zman as service. Running zman as a service enables you to execute zman commands in service mode, which reduces the execution time for the commands.

The zman service commands are grouped within the shell category and are as follows:

zman shell-start-as-service (ssas) [options]

Starts the zman service.

Accepts the following options:

-s or --servicePort: Specify the port for the zman server to run. The default port is 2020

zman shell-stop-service (sss)

Stops the zman service.

zman shell-status (ssts)

Displays the zman service status.

To run zman commands as a service, you must do the following:

- Before starting the zman service, run the `zman admin-store-credential(asc)` command to save the credentials. The zman service uses the saved credentials to run the zman commands in service mode.

- If you choose to run any zman command that requires file paths in service mode, you must specify the complete file path instead of the relative file path. By default, the working directory is initialized to the location where the zman service is started.

The following zman commands run as separate processes even when the zman service is up and running:

```
zman shell-stop-service(sss)
zman admin-create(ac)
zman admin-store-credential(asc)
zman certificate-authority-export(cae)
zman certificate-authority-import(cai)
zman zenserver-backup(zsb)
zman zenserver-restore(zsr)
```

List of Commands

Asset Management Commands

These commands are used to load purchase record files and refresh the compliance engine. They apply only to ZENworks 11 SP2 Asset Management.

asset-management-import-contract (amic) [options]

Imports a contract to a file. The command queues the import task to be processed by the loader. The loader then loads the contracts from the file.

The command is asynchronous, which means that zman returns immediately and does not wait for the import to be complete. This is because the command might run for a long time. You can check the progress of the import by viewing the log file created by the command.

Accepts the following option:

-f, --filename=[File path]: Specify the full path of the comma-separated (.csv or .txt) file to be imported.

asset-management-load-purchase-record (amlpr)

Loads a purchase record file. The Purchase Record Loader looks for the purchase record file to load on the primary server in the folder `ZENWORKS_HOME/temp/PurchaseRecordImport/uploaded`. If you have not yet used the Purchase Record Loader from ZENworks Control Center, you need to create this folder.

This command is asynchronous, which means `zman` returns immediately and does not wait for the command to complete. This is because the command may run for a long time. You can check the status of this process from ZENworks Control Center.

asset-management-refresh-compliance-data (amrcd)

Runs the software compliance engine to refresh the software license compliance data.

This command is asynchronous, which means zman returns immediately and does not wait for the command to complete. This is because the command may run for a long time. You can check the status of this process from ZENworks Control Center.

Administrator Commands

These commands are used to create and manage administrator accounts. Administrator commands begin with `admin-` in the long form or the letter `a` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

- "admin-clear-credential (acc)" on page 13
- "admin-create (ac) (administrator or user) [options]" on page 13
- "admin-delete (ad) (ZENworks administrator name) [...]" on page 14
- "admin-list (al) [options]" on page 14
- "admin-list-roles (alr)(ZENworks administrator name) [options]" on page 14
- "admin-rename (ar) (ZENworks administrator name) (new name)" on page 14
- "admin-rights-delete (ard) (ZENworks administrator name) [options] [object path]" on page 14
- "admin-rights-info (ari) (ZENworks administrator name) [object path] [options]" on page 14
- "admin-rights-set (ars) [ZENworks administrator name] [rights category] [object path] [...] [options]" on page 15
- "admin-store-credential (asc) (ZENworks administrator name)" on page 15

admin-clear-credential (acc)

Clears the ZENworks username and password stored by using the `admin-store-credential` command. It is always safer and is a good practice to delete the stored credentials after you are finished executing the commands.

admin-create (ac) (administrator or user) [options]

Creates a ZENworks administrator account.

(administrator or user) - Name of the administrator or full path of a user in a user source.

Accepts the following options:

-f, --fullname=[full name] - Full name of the ZENworks administrator.

-a, --assign-same-rights - Gives the new administrator the same rights as the administrator who executes this command.

The command prompts for the password of the administrator, to avoid entering the password in clear text. If the administrator account is based on a user, it uses the same credentials defined in the user source so the command does not prompt for the password.

admin-delete (ad) (ZENworks administrator name) [...]

Deletes a ZENworks administrator account.

(ZENworks administrator name) [...] - Name of the ZENworks administrator. The wildcard * can be used in the object names if it is enclosed in quotation marks. Exercise caution while using wildcards for deleting objects.

admin-list (al) [options]

Lists all ZENworks administrator accounts.

Accepts the following options:

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

admin-list-roles (alr)(ZENworks administrator name) [options]

Lists roles assigned to a ZENworks administrator account.

Accepts the following option:

-l, --longnames - Displays names for the rights. By default, the abbreviated form of the rights name is displayed.

admin-rename (ar) (ZENworks administrator name) (new name)

Renames a ZENworks administrator account.

admin-rights-delete (ard) (ZENworks administrator name) [options] [object path]

Deletes rights assigned to a ZENworks administrator account.

[object path] [...] - Full path of the ZENworks object, starting with a forward slash (/).

Accepts the following option:

-c, --category=[rights category] - Name of the category within which the rights are grouped. To see the list of categories, run zman admin-rights-set --Help | more.

admin-rights-info (ari) (ZENworks administrator name) [object path] [options]

Displays rights assigned to a ZENworks administrator account.

[object path] [...] - Full path of the ZENworks object, starting with a forward slash (/).

Accepts the following options:

-c, --category=[rights category] - Name of the category within which the rights are grouped. To see the list of categories, run zman admin-rights-set --Help | more.

-e, --effective - Displays the effective rights of the administrator. Effective rights include rights inherited from roles assigned to the administrator. If this option is not specified, rights assigned directly to the administrator are displayed.

-l, --longnames - Displays names for the rights. By default, the abbreviated form of the rights name is displayed.

`admin-rights-set (ars) [ZENworks administrator name] [rights category] [object path] [...] [options]`

Allows or denies rights for a ZENworks administrator account.

[rights category] - Name of the category under which the rights are grouped. To see the list of categories, run `zman admin-rights-set --Help | more`.

[object path] [...] - Full path of the ZENworks object, starting with forward slash (/), on which the rights should be enforced. To view the root folders of the allowed object types for which rights can be assigned within each rights category, run `zman admin-rights-modify --Help | more`.

Accepts the following options:

-a, --allow=[right][...] - A comma-separated list of long or short names of the rights to be allowed. To view the rights for each category, run `zman admin-rights-set --Help | more`.

-A, --allow-all - Allows all rights of the Rights category.

-d, --deny=[right][...] - A comma-separated list of long or short names of the rights to be denied. To view the rights of each category, run `zman admin-rights-set --Help | more`.

-D, --deny-all - Denies all rights of the Rights category.

-f, --folder=[folder name] - Full path of a folder. If specified, the path of the objects is determined relative to this folder. This can be used to specify multiple objects under the same folder.

-H, --Help - Displays detailed help about the categories of rights available, and the long and short form of rights names within each category. If a category name is specified, help is provided for that category.

`admin-store-credential (asc) (ZENworks administrator name)`

Stores the ZENworks username and password for use with every command so credentials need not be entered for every command. The credentials are stored on the home directory of the logged-in user and are used only for that logged-in user. In Linux, ensure that the user has a home directory. If you share the same machine login with other ZENworks administrators, you must clear your credentials by using the `admin-clear-credential` command after you are finished executing commands.

Batch Command

The batch command is a utility command used to execute a list of zman commands from a text file. The commands are executed faster because logging into ZENworks is done only once and not for every command. Batch commands begin with `batch-` in the long form, or with the letter `b` in the short form.

This command accepts the option flags listed in the Global Options section. In addition, it accepts individual options as listed with the command.

- ◆ "batch-execute (bex) (file path) [...] [options]" on page 15

`batch-execute (bex) (file path) [...] [options]`

Executes zman commands from text files in a batch (not to be confused with Windows batch files).

(file path) [...] - The path of the text files containing zman commands to be executed. Lines starting with #, `rem`, `REM`, ; and // are considered as comments, and are not processed. For a sample file, refer to `BatchExecuteCommands.txt` located in `/opt/novell/zenworks/share/zman/samples/batch` on a Linux server or `Installation_path:\Novell\Zenworks\share\zman\samples\batch` on a Windows server.

Accepts the following option:

-e, --exit-on-failure - Stops executing the commands and exits if a command fails partially or totally. A command is said to partially fail for commands that take multiple arguments when the processing of one or more of the arguments was not successful. For example, for `bundle-delete`, the command is said to partially fail if one of the arguments could not be found or deleted. By default, the commands in the file are executed continuously without checking for failure.

Bundle Commands

These commands are used to create, modify and manage bundles. Bundle commands begin with `bundle-` in the long form, or with the letter `b` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command. These commands apply only to ZENworks 11 SP2 Configuration Management.

bundle-add-actions (baa) (bundle path) (action XML file) [...] [options]

Adds actions to a bundle.

(action XML file) [...] - The XML files that contain information for actions to be added. The actions are grouped under one of these action sets: Distribution, Install, Launch, Verify, Uninstall, Terminate. The XML file contains an `ActionSet` element that contains information of actions to be added for an action set. Multiple XML files can be given as input to add actions to different action sets of the bundle. The XML files can be created by exporting actions of an existing bundle using the `bundle-export-actions` command.

Accepts the following option:

-a, --actioninfo=[content and dependency for actions] - XML file containing content and bundle dependency information for actions that have file content or dependency on another bundle. For example, for `Install MSI Action`, the MSI file to be installed is the file content. `Install Bundle Action` takes another bundle as dependency so this data also needs to be specified for proper functioning of these actions. For the XML format template, refer to `ActionInfo.xml` located in `/opt/novell/zenworks/share/zman/samples/bundles` on a Linux server or `Installation_directory`:`\Novell\Zenworks\share\zman\samples\` `bundles` on a Windows server.

NOTE: If you choose the action set type as Uninstall, the following options are not supported for Linux bundles and Linux Patch bundles: DaysBeforeUninstallWhenNotUsed, DisplayUninstallMessage, IgnoreUninstallAfterUnassociation, IgnoreChainedDependencies, and AllowToCancelUninstall.

```
bundle-assign (ba) (device or user type) (bundle or bundle group) (device or user
object path) [...] [options]
```

Assigns a bundle or bundle group to one or more device or user objects.

(device or user type) - Valid values are `device`, `server`, `workstation`, and `user`.

(device or user object path) [...] - The path of the device or user objects relative to the root folder of the device or user type specified.

Accepts the following options:

-f, --folder=[bundle folder] - The path of a bundle folder relative to `/Bundles`. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

-I, --icon-location=[application location XML file] - XML file that contains the locations to place the icon for the bundle application. For the XML file format, refer to `IconLocation.xml` located in `/opt/novell/zenworks/share/zman/samples/bundles` on a Linux server or *Installation_directory*`:\Novell\Zenworks\share\zman\samples\` `bundles` on a Windows server.

-d, --distribution-schedule=[distribution schedule XML file] - XML file that contains the distribution schedule.

-l, --launch-schedule=[launch schedule XML file] - XML file that contains the launch schedule.

-a, --availability-schedule=[availability schedule XML file] - XML file that contains the availability schedule. For the schedule XML file templates, refer to the XML files located in `/opt/novell/zenworks/share/zman/samples/schedules` on a Linux server or *Installation_directory*`:\Novell\Zenworks\share\zman\samples\` `schedules` on a Windows server.

-i, --install-immediately - Installs the bundle immediately after distribution.

To use this option, you must also specify the Distribution schedule. The Distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-L, --launch-immediately - Launches the bundle immediately after installation.

To use this option, you must also specify the Distribution schedule. The Distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-n, --distribute-now - Sets the distribution schedule to distribute the bundle immediately. If this option is specified, the --distribution-schedule and --distribute-on-device-refresh options are ignored. The --distribute-now, --distribute-on-device-refresh, and --distribution-schedule options are mutually exclusive and are used to set the distribution schedule. The --distribute-now option is considered first, followed by --distribute-on-device-refresh and --distribution-schedule.

-r, --distribute-on-device-refresh - Sets the distribution schedule to distribute the bundle on device refresh. If this option is specified, the --distribution-schedule option is ignored.

-s, --launch-on-device-refresh - Sets the launch schedule to launch the bundle on device refresh. If this option is specified, the --launch-schedule option is ignored.

-w, --wakeup-device-on-distribution - Wakes up the device by using Wake-On-LAN if it is shut down while distributing the bundle.

To use this option, you must also specify the Distribution schedule. The Distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-B, --broadcast=[Broadcast address][...] - A comma-separated list of addresses used to broadcast the Wake-On-LAN magic packets. This option is used only if you choose to wake up the device by using Wake-On-LAN. A valid IP address is a valid value.

-S, --server=[Path of the Primary or Proxy Server objects relative to `/Devices`][...] - A comma-separated list of Primary or Proxy Server objects used to wake up the device. This option is used only if you choose to wake up the device by using Wake-On-LAN.

-C, --retries=[Number of retries] - Number of times the Wake-On-LAN magic packets are sent to the device(s). This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 0 and 5. The default value is 1.

-T, --timeout=[Time interval between retries] - The time interval between two retries. This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 2 and 10 (in minutes). The default value is 2.

bundle-copy (bcp) (bundle path) (bundle name) [parent folder] [options]

Copies a bundle.

Accepts the following options:

-d, --desc=[description] - Description for the bundle.

-i, --icon-file=[bundle icon file] - The path of the image file containing the image that should be used as the bundle icon. If this option is not specified, the contents of the subfolders are not listed.

bundle-create (bc) (bundle name) (bundle XML file) [parent folder] [options]

Creates a new bundle.

(bundle XML file) - XML file containing exported bundle information. Use `bundle-export-to-file (betf)` to export a existing bundle's information into an XML file. If you want to reuse files, template XML files can be created from bundles that were created through ZENworks Control Center. For a sample XML file, refer to `WindowsMSIBundle.xml` located in `/opt/novell/zenworks/share/zman/samples/bundles` on a Linux server or `Installation_directory`:\Novell\Zenworks\share\zman\samples\bundles on a Windows server.

Accepts the following options:

-d, --desc=[description] - Description for the bundle.

-a, --actioninfo=[content and dependency for actions] - XML file containing content and bundle dependency information for actions that have file content or dependency on another bundle. For example, for `Install MSI Action`, the MSI file to be installed is the file content. `Install Bundle Action` takes another bundle as a dependency. This data also needs to be specified for proper functioning of these actions. For the XML format template, refer to `ActionInfo.xml` located in `/opt/novell/zenworks/share/zman/samples/bundles` on a Linux server or `Installation_directory`:\Novell\Zenworks\share\zman\samples\bundles on a Windows server.

-i, --icon-file=[bundle icon file] - The path of the image file containing the image that should be used as the bundle icon.

-s, --create-as-sandbox=[create as sandbox] - Creates the bundle as a sandbox.

-x, --actions-as-in-xml - Creates default actions for the new bundle as found in the template XML file. For example if you delete Verify Install default action (for the Verify action type) from the bundle and export the bundle information into an XML file, all default actions found in the XML file excluding the Verify Install default action will be created for the new bundle. If you do not specify this option, a bundle will be created with all the default actions including the Verify Install default action which is not found in the XML file.

NOTE: If you choose the action set type as Uninstall, the following options are not supported for Linux bundles and Linux Patch bundles: DaysBeforeUninstallWhenNotUsed, DisplayUninstallMessage, IgnoreUninstallAfterUnassociation, IgnoreChainedDependencies, and AllowToCancelUninstall.

---

bundle-delete (bd (bundle object path) [...][options]

Deletes one or more bundle objects.

(bundle object path) [...] - The path of the bundle objects (bundle, bundle folder or bundle group) relative to /Bundles. The wildcard * can be used in the object names if it is enclosed in quotation marks. Exercise caution while using wildcards for deleting objects.

Accepts the following options:

-r, --recursive - Deletes objects inside a folder recursively.

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

bundle-delete-version (bdv) (bundle path) [bundle version number] [options]

Deletes one or more versions of the bundle.

[bundle version number] [...] - Version of the bundle to delete.

Accepts the following option:

-a --all - Deletes all the older versions of the bundle.

-p, --previous - Deletes all the versions of the bundle older than the specified version.

Examples:

To delete the version 5 of the bundle, zenbundle:

 zman bdv zenbundle 5

To delete all the versions of the bundle, zenbundle:

 zman bdv zenbundle -a

To delete all the versions of the bundle, zenbundle, that are older than the version 3:

 zman bdv zenbundle version -p 3

bundle-disable (bdl) (bundle path) [...] [options]

Disables bundles.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

bundle-enable (bel) (bundle path) [...] [options]

Enables bundles.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

bundle-export-actions (bea) (bundle path) (XML file path) (action set type) [action's positions] [...]

Exports the actions added to a bundle's action set to a file. The file can be used as input to the bundle-add-actions command.

(action set type) - The type of the action set. If this option is specified, the actions for only this action set are listed. Valid values are Install, Launch, Verify, Uninstall, Terminate, and Preboot.

NOTE: The Terminate action set is not supported for Linux bundles and Linux Patch bundles.

[action's positions] [...] - A comma-separated list of position of actions in the action set. The first action is at position 1. Use the `bundle-list-actions` command to see the position of a particular action in an action set. This is optional, but if it is not specified, all the actions in the specified action set are exported.

bundle-export-to-file (betf) (bundle path) (XML file path) [version of the bundle][-c|--export-content]

Exports a bundle's information (in XML format) to a file. The XML file is to be used as input for creating bundles.

(XML file path) - The complete path of the XML file to which the bundle's information is to be exported.

[version of the bundle] - Version of the bundle to be exported. If it not specified, the published version of the bundle is considered. To export a sandbox version of the bundle, specify sandbox.

[-c|--export-content] - Exports the content of the bundle to a sub directory within the directory containing the XML file to which the bundle's information is exported. If the option is not specified, the bundle content is not exported.

Examples:

To export the sandbox version of a bundle named zenbundle to an xml file named `bundle.xml`:

`zman betf bundlefolder/zenbundle C:\bundles\bundle.xml sandbox`

To export the version 3 of a bundle named zenbundle to an xml file named `bundle.xml`:

`zman betf bundlefolder/zenbundle C:\bundles\bundle.xml 3`

To export the version 5 of a bundle named zenbundle to an xml file named `bundle.xml` and export the bundle's content to the `C:\bundles\bundle_content` directory:

`zman betf bundlefolder/zenbundle C:\bundles\bundle.xml 5 -c`

bundle-folder-create (bfc) (folder name) [parent folder] [options]

Creates a new folder for containing bundles.

Accepts the following option:

--desc=[description] - Description for the folder.

bundle-group-add (bga) (bundle group path) (bundle path) [...] [options]

Adds bundles to a bundle group.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

bundle-group-create (bgc) (group name) [parent folder] [options]

Creates a bundle group and adds members to it.

Accepts the following options:

--desc=[description] - Description for the group.
-m, --members=[bundle path][...] - The path of the bundles relative to /Bundles.
-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

`bundle-group-members (bgm) (bundle group path) [options]`

Lists members of a bundle group.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`bundle-group-remove (bgr) (bundle group path) (bundle path) [...] [options]`

Removes bundles from a bundle group.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

`bundle-increment-version (biv) bundle path) [...] [options]`

Increments the version of bundles. This command is applicable for versions prior to ZENworks 11. In ZENworks 11 or higher, this command is replaced by the `bundle-sandbox-publish (bsp)` command

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

`bundle-list (bl) [options] [folder]`

Lists bundles objects.

Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, the contents of the subfolders are not listed.

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.

-c, --count - Displays the count of the results.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`bundle-list-actions (bla) (bundle path) [action set type]`

Lists actions added to a bundle.

[action set type] - The type of the action set. If this option is specified, the actions for only this action set are listed. Valid values are `Install`, `Launch`, `Verify`, `Uninstall`, `Terminate`, and `Preboot`.

NOTE: The Terminate action set is not supported for Linux bundles and Linux Patch bundles.

`bundle-list-assignments (blas) (bundle or bundle group) [options]`

Lists the device and user assignments for a bundle.

Accepts the following options:

-t, --typefilter=[assignment type] - Filters on the assignment type. Valid values are `device` and `user`.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`bundle-list-groups (blg) (bundle path) [options]`

Lists the bundle groups of which the given bundle is a member.

Accepts the following options:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

Examples:

To list all the bundle groups that bundle1 belongs to with five results displayed at a time:

```
zman blg bundlefolder/bundle1 -s 5
```

bundle-list-version (blv) (bundle path)

Lists all the versions of the bundle.

Examples:

To list all the versions of the bundle, zenbundle:

```
zman blv zenbundle
```

bundle-move (bmv) (bundle object path) [destination folder path]

Moves a bundle object to a different folder.

bundle-remove-actions (bra) (bundle path) (action set type) [action's positions]
[...]

Removes actions from a bundle.

(action set type) - The type of the action set that contains the action. Valid values are Install, Launch, Verify, Uninstall, Terminate, and Preboot.

NOTE: The Terminate action set is not supported for Linux bundles and Linux Patch bundles.

[action's positions] [...] - A comma-separated list of position of actions in the action set. The first action is at position 1. Use the bundle-list-actions command to see the position of a particular action in an action set. This is optional, but if it is not specified, all the actions in the specified action set are removed.

bundle-rename (br) (bundle object path) (new name) [options]

Renames a bundle object.

-p, --publish =[publish] - If this option is specified and the display name of the bundle is edited, the bundle is immediately published. However, if the current display name of the bundle is different from the existing bundle name or if a sandbox already exists for the bundle, specifying this option will have no effect.

bundle-reorder-actions (broa) (bundle path) (action set type) (current position)
(new position)

Reorders actions in the action set of a bundle.

(action set type) - The type of the action set that contains the action. Valid values are Install, Launch, Verify, Uninstall, Terminate, and Preboot.

NOTE: The Terminate action set is not supported for Linux bundles and Linux Patch bundles.

---

bundle-sandbox-create (bsc) (bundle path) [version of the bundle] [options]

Creates a sandbox from an existing version of a bundle

[version of the bundle] - Specify the version of the bundle to be used for creating the sandbox.

Accepts the following option:

-f, --force =[force] - Overwrites any existing sandbox.

Example:

To create a sandbox from the version 2 of a bundle, zenbundle:

```
zman bsc zenbundle 2
```

bundle-sandbox-publish (bsp) (bundle path) [options]

Publishes a sandbox to create a published version.

Accepts the following option:

-f, --force =[force] - If a primary bundle has dependent child bundles with the sandbox only version, you must specify this option to publish the changes made both to the primary and the dependent bundles. If you do not specify this option, the publish of the primary bundle also fails.

-c, --allChild =[all children] - If a primary bundle has dependent child bundles with the sandbox version, you can specify this option to publish the changes made both to the primary and the dependent child bundles.

-i, --incAllParent =[increment all parents] - If a primary bundle has parent bundles, you can specify this option to increment the version of the parent bundle to apply the newly published changes made to the primary bundle to the devices that are assigned to the parent bundles.

-p, -forceParent =[force parent] - If a primary bundle has parent bundles and you choose to increment the version of the parent bundle, then you must specify this option to publish any of the parent bundles which already have a sandbox version.

Examples:

To publish the sandbox version of a bundle, zenbundle:

```
zman bsp zenbundle
```

To publish the sandbox version of a bundle named zenbundle1 along with all its dependent bundles:

```
zman bsp zenbundle1 -f -c
```

bundle-sandbox-publish-to-newbundle (bsptn) (bundle path) (bundle name) [parent folder] [options]

Publishes a sandbox to a new bundle.

Accepts the following option:

-s, --createAsSandbox =[create as sandbox] - Create the bundle as a sandbox.

-g, --groups - Adds the newly created bundle to all the groups which the primary bundle is member of.

-d, --deviceAssignments - Copies all the device assignments from the primary bundle to the new bundle.

-u, --userAssignments - Copies all the user assignments from the primary bundle to the new bundle.

-f, --force =[force] - If a primary bundle has dependent child bundles with the sandbox only version, you must specify this option to publish the changes made both to the primary and the dependent bundles. If you do not specify this option, the publish of the primary bundle also fails.

-c, --allChild =[all children] - If a primary bundle has dependent child bundles with the sandbox version, you can specify this option to publish the changes made both to the primary and the dependent child bundles.

-i, --incAllParent =[increment all parents] - If a primary bundle has parent bundles, you can specify this option to increment the version of the parent bundle to apply the newly published changes made to the primary bundle to the devices that are assigned to the parent bundles.

-p, -forceParent =[force parent] - If a primary bundle has parent bundles and you choose to increment the version of the parent bundle, then you must specify this option to publish any of the parent bundles which already have a sandbox version.

-n, -depsToNewFolder =[publish Dependents bundles to new folder] -Path of the bundle folder in which the dependent bundles are published as new bundles. This folder should not contain any of the dependent bundle selected to be published because the new bundle is published with the same name as the original dependent bundle.

Examples:

To publish the sandbox version of a bundle named zenbundle1 as a new bundle named zenbundle2:

```
zman bsptn zenbundle1 zenbundle2 /Bundles/Folder1
```

To publish the sandbox version of a bundle named zenbundle1 as a new bundle named zenbundle2 and publish the dependent bundles of zenbundle1 as new bundles:

```
zman bsptn zenbundle1 zenbundle2 /Bundles/Folder1 depsToNewFolder=/Bundles/
Folder2
```

bundle-sandbox-revert (bsr) (bundle path)

Deletes the sandbox and reverts to the latest version of the bundle.

Example:

To delete the sandbox of a bundle, zenbundle and revert to the latest version of the bundle:

```
 zman bsr zenbundle
```

bundle-settings-copy (bscp) (source bundle or bundle folder path) (destination bundle or bundle folder path) [...] [options]

Copies settings.

(source bundle or bundle folder path) - The path of the bundle or bundle folder relative to /Bundles from which settings have to be copied.

(destination bundle or bundle folder path) [...] - The path of the bundles or bundle folders relative to /Bundles to which settings must be copied.

Accepts the following option:

-n, --names=[settings name][...] - Names of the settings to be copied from the source path. If this option is not specified, all settings defined at the source path are copied.

bundle-settings-export-to-file (bsetf) [options] (XML file path) [settings name] [...]

Exports settings data (in XML format) to a file. The XML file is to be used as input for creating or modifying settings.

(XML file path) - The file into which the settings data is stored in XML format. If the file does not exist, a new file is created.

[settings name] [...] - Names of the settings to be exported. If this option is not specified, all settings are exported.

Accepts the following options:

-s, --source=[source bundle or bundle folder path] - The path of the bundle or bundle folder relative to /Bundles from which settings must be exported. If this option is not specified, settings are exported from the Zone.

-e, --effective - If specified, the effective settings are retrieved; otherwise, only the settings defined or overridden at the source path are retrieved.

bundle-settings-revert (bsrt) (source bundle or bundle folder path) (settings name) [...]

Reverts the settings to that of the parent folder.

(source bundle or bundle folder path) - The path of the bundle or bundle folder relative to /Bundles, whose settings must be reverted.

bundle-settings-set (bss) (XML file path) [destination bundle or bundle folder path] [...]

Sets settings at various levels.

(XML file path) - XML file containing exported settings information. Use settings-export-to-file (setf) to export settings information into a XML file.

Accepts the following options:

[settings name] [...] - Names of the settings to be exported. If this option is not specified, all settings are exported.

-f, --force - Forces all children (subfolders and individual bundles) to inherit these settings.

-s, --source=[source bundle or bundle folder path] - The path of the bundle or bundle folder relative to /Bundles for which settings have to be set. If this option is not specified, settings are set at the Zone level.

bundle-unassign (bua) (device or user type) (bundle or bundle group) (device or user object path) [...] [options]

Unassigns a bundle or bundle group from one or more device or user objects.

(device or user type) - Valid values are device, server, workstation, and user.

(device or user object path) [...] - The path of the device or user objects relative to the root folder of the device or user type specified.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

bundle-update-assignment (bupa) (device or user type) (bundle or bundle group) (device or user object path) [...] [options]

Updates the assignment between a bundle or bundle group and device or user objects.

(device or user type) - Valid values are device, server, workstation, and user.

(device or user object path) [...] - The path of the device or user objects relative to the root folder of the device or user type specified.

Accepts the following options:

-f, --folder=[device or user folder] - The path of the device or user folder relative to the respective root folder. If this option is specified, the path of the device or user objects is determined relative to this folder. This can be used to specify multiple device or user objects with the same folder.

-I, --icon-location=[application location XML file] - XML file that contains the locations to place the icon for the bundle application. For the XML file format, refer to `IconLocation.xml` located in `/opt/novell/zenworks/share/zman/samples/bundles` on a Linux server or *`Installation_directory`*`:\Novell\Zenworks\share\zman\samples\bundles` on a Windows server.

-d, --distribution-schedule=[distribution schedule XML or `NoSchedule`] - XML file that contains the distribution schedule or `NoSchedule` if the schedule must be removed.

-l, --launch-schedule=[launch schedule XML file or `NoSchedule`] - XML file that contains the launch schedule or `NoSchedule` if the schedule must be removed.

-a, --availability-schedule=[availability schedule XML file or `NoSchedule`] - XML file that contains the availability schedule or `NoSchedule` if the schedule must be removed. For the schedule XML file templates, refer to the XML files located in `/opt/novell/zenworks/share/zman/samples/schedules` on a Linux server or *`Installation_directory`*`:\Novell\Zenworks\share\zman\samples\schedules` on a Windows server.

-i, --install-immediately=[yes or no] - Install the bundle immediately after distribution, or remove the same option set during bundle assignment. Valid values are `true` or `yes`, and `false` or `no`.

-L, --launch-immediately=[yes or no] - Launch the bundle immediately after installation, or remove the same option set during bundle assignment. Valid values are `true` or `yes`, and `false` or `no`.

-n, --distribute-now=[yes or no] - Sets the distribution schedule to distribute the bundle immediately. If this option is specified, the --distribution-schedule and --distribute-on-device-refresh options are ignored. The --distribute-now, --distribute-on-device-refresh, and --distribution-schedule options are mutually exclusive and are used to set the distribution schedule. The --distribute-now option is considered first, followed by --distribute-on-device-refresh and --distribution-schedule.

-r, --distribute-on-device-refresh=[yes or no] - Sets the distribution schedule to distribute the bundle on device refresh, or removes the same option set during bundle assignment. Valid values are `true` or `yes`, and `false` or `no`. If this option has the value `true` or `yes`, the --distribution-schedule option is ignored and any previously set distribution schedule is overwritten.

-s, --launch-on-device-refresh=[yes or no] - Sets the launch schedule to launch the bundle on device refresh. If this option is specified, the --launch-schedule option is ignored.

-w, --wakeup-device-on-distribution=[yes or no] - Wakes up the device by using Wake-On-LAN if it is shut down while distributing the bundle, or removes the same option set during bundle assignment. Valid values are `true` or `yes`, and `false` or `no`.

-B, --broadcast=[Broadcast address][...] - A comma-separated list of addresses used to broadcast the Wake-On-LAN magic packets. This option is used only if you choose to wake up the device by using Wake-On-LAN. A valid IP address is a valid value.

-S, --server=[Path of the Primary or Proxy Server objects relative to `/Devices`][...] - A comma-separated list of Primary or Proxy Server objects used to wake up the device. This option is used only if you choose to wake up the device by using Wake-On-LAN.

-C, --retries=[Number of retries] - Number of times the Wake-On-LAN magic packets are sent to the devices. This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 0 and 5. The default value is 1.

-T, --timeout=[Time interval between retries] - The time interval between two retries. This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 2 and 10 (in minutes). The default value is 2.

bundle-view-advanced-deployment-status (bvads) (bundle path) [options]

Displays advanced deployment status of a bundle.

Accepts the following options:

-d, --device=[device path] - Displays the deployment status only for the specified device. The path of the device is relative to /Devices.

-u, --user=[user path] - Displays the deployment status only for the specified user. The path of the user is relative to /Users. If device is also specified, status details for the specified user logged into the specified device are displayed.

-n, --namefilter=[target device name] - Filters on the name of the device. Displays options matching the specified filter. The wildcards * and ? can be used if they are enclosed in quotations.

--statusfilter=[status type][...] - Filters on the status of Bundle Distribution and Install Events. Valid values are S, F, and P (Success, Failure, and Pending). A comma-separated list of status types can be given.

-t, --typefilter=[target device or user type][...] - Filters on the type of the target. Valid values are `server`, `workstation`, and `user`. A comma-separated list of target type can be given.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

bundle-view-assignment-schedule (bvas) (device or user type) (bundle or bundle group) (device or user object path)

Displays the deployment schedules for a bundle assigned to a device or user object.

(device or user type) - Valid values are `device`, `server`, `workstation`, and `user`.

(device or user object path) [...] - The path of the device or user objects relative to which the bundle group is assigned.

bundle-view-summary-totals (bvst) (bundle path)

Displays a summary of the deployment status of a bundle.

linux-export-actions (lea) (Action Type) (Action XML File path) (Content Files location Path) [options]

Creates an XML file to export package metadata. Use the `baa` command to add the exported content to the server.

(Action Type) - Specify the action for which you want to create the XML file. The supported action types are `Install RPM` and `Distribute RPM`.

(Action XML File path) - Location of the generated Action Definition XML file.

(Content Files location Path) - Location of the RPM content files on the local machine.

Accepts the following option:

-t, --target - Specify a valid linux distribution target for packages.

Certificate Authority Commands

These commands are used to manage the certificate authority role of ZENworks servers. Certificate authority commands begin with `certificate-authority-` in the long form, or with the letters `ca` in the short form.

`certificate-authority-export (certificate-authority-export/cae) (file path) [options]`

Exports the key-pair credentials of the zone certificate authority to a file, and optionally disables the Certificate Authority role of the local server.

Accepts the following option:

-d, --disable-CA-role - Removes the Certificate Authority role of the local server.

`certificate-authority-import (certificate-authority-import/cai) (file path)`

Imports the key-pair credentials of the zone certificate authority from a file and enables the Certificate Authority role on the local server.

`certificate-authority-role-disable (card)`

Disables the Certificate Authority role on the local server.

`certificate-authority-role-enable (care)`

Enables the Certificate Authority role on the local server.

Credential Commands

These commands are used to manage the credentials needed for certain ZENworks actions and tasks that require authentication to resources. Credential commands begin with `credentials-` in the long form, or with the letters `cr` in the short form.

`credentials-create (crc) (credentialName [folder]) (--username) [--password] [--desc]`

Creates credentials.

Accepts the following options:

-u, --user - The username to access the resource.
--password - The password used to access the resource.
-d, --desc - A description of the credential.

`credentials-delete (crd) (credentialPath)`

Deletes credentials.

`credentials-folder-create (crf) (folder name) [parent folder] [options]`

Creates a new folder for containing credentials.

Accepts the following option:

--desc=[description] - Description for the folder.

credentials-list (crl) [folder]

Lists credentials.

credentials-move (crm) (credential name) [destination folder path]

Moves a credential.

credentials-rename (crr) (credentialPath) (newcredentialPath)

Renames credentials.

credentials-update (cru) (--username) [--password] [--desc]

Creates credentials.

Accepts the following options:

-u, --user - The username to access the resource.

--password - The password used to access the resource.

-d, --desc - A description of the credential.

Database Commands

These commands are used to manage the embedded Sybase SQL Anywhere database. Database commands begin with database- in the long form, or with the letter d in the short form.

NOTE: These commands cannot be used to manage an external Sybase SQL Anywhere database.

database-backup (db) (backup directory) [schedule SQL file] [options]

Backs up the embedded Sybase SQL Anywhere database and also allows you to schedule the backup operation.

(backup directory) - The local directory on the database server or the network location where the database files are backed up. Ensure that the parent directory already exists and has sufficient disk space. Any existing database files in the directory are overwritten.

[schedule SQL file] - File that contains the schedule for backing up the database. The SQL file can contain CREATE EVENT, ALTER EVENT, or DROP EVENT SQL statements. For sample SQL files, refer to the files located in /opt/novell/zenworks/share/zman/samples/database on a Linux server or
Installation_directory:\Novell\Zenworks\share\zman\samples\database on a Windows server. If you do not specify a schedule file, the database is backed up immediately. For more information on defining schedules, refer to the SQL Anywhere documentation at http:// /infocenter.sybase.com/help/index.jsp?topic=/com.sybase.dbdaen10/html/title.htm.

Accepts the following option:

-d, --dir-name=[SQL function call] - SQL function call that returns a value. The value is appended to the backup directory path. For example, if this command runs on Tuesday with the backup directory specified as c:\ and the value for this option specified as DAYNAME(now()), the files are backed up to c:\Tuesday.

database-get-credentials (dgc)

Retrieves the credentials used to connect to the embedded Sybase SQL Anywhere database.

Deployment Commands

These commands are used to perform deployment tasks. Deployment commands begin with `deployment-` in the long form, or with the letters `dp` in the short form.

`deployment-task-abort (dpta) (deployment task name) [...]`

Aborts the deployment task.

`deployment-task-create (dptc) (deployment task name) (user credential file) [IP address/DNS name] [...] [options]`

Creates a deployment task to deploy the ZENworks Adaptive Agent to devices by using the IP address or DNS name.

(user credential file) - The path of the file containing the credentials to connect to the device. Each line in the file should have a username and password separated by a space. For example, administrator password.

Accepts the following options:

-f, --file=[IP address file][...] - A comma-separated list of file paths containing the IP addresses or DNS names of devices to which the ZENworks Adaptive Agent should be deployed. The file can contain a list of IP addresses or DNS names of devices in comma-separated-value (CSV) format, or one IP address or DNS name on each line.

-s, --schedule=[launch schedule XML file] - XML file that contains the launch schedule. For the schedule XML file templates, refer to the XML files located in `/opt/novell/zenworks/share/zman/samples/schedules` on a Linux server or `Installation_directory:\Novell\Zenworks\share\zman\samples\schedules` on a Windows server. If the schedule file or the `--run-now` option is not specified, the task is not scheduled to run.

-r, --run-now - Schedules the deployment task to run immediately after creation.

-b, --reboot=[Reboot option] - Indicates when the device must be rebooted after deploying the agent. Valid values are `immediate` and `manual`.

-d, --desc=[description] - Description for the deployment task.

-p, --proxy=[Windows Proxy] - The path of a Windows device relative to `/Devices`. In order to deploy management agents to Microsoft Windows devices from a Linux Primary Server, a managed device running Microsoft Windows is needed. Ignore this option if you are running the deployment task from a Windows Primary Server.

-t, --timeout=[Time out] - The number of seconds you want the Primary Server to wait for a response from the Windows Proxy.

-k, --key=[Registration key] - Registration key for registering the device.

`deployment-task-delete (dptd) (deployment task name) [...]`

Deletes the deployment task.

`deployment-task-list (dptl) [options]`

Lists the deployment tasks and the status.

Accepts the following options:

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards *
and ? can be used if they are enclosed in quotations marks.

-c, --count - Displays the count of the results.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`deployment-task-run-now (dptrn) (deployment task name) [...]`

Schedules the specified deployment task to run immediately. The task should have the
credentials saved in the data store.

Discovery Commands

These commands are used to view device discovery-related information. Discovery commands begin
with `discovery-` in the long form, or with the letter `d` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition,
they accept individual options as listed with each command.

- "discovery-view-discovered-devices (dvdd) [options]" on page 32
- "discovery-task-list (dtl) [options]" on page 32
- "discovery-task-run-now (dtrn) (discovery task name)" on page 32

`discovery-view-discovered-devices (dvdd) [options]`

Lists the discovered devices.

Accepts the following options:

-n, --nameFilter=[filter string] - Displays the devices matching the specified filter. The wildcard *
can be used if it is enclosed in quotations.

-t, --typefilter=[type filter] - Displays options matching the specified type. Valid values are
`server`, `workstation`, `printer`, `network`, `thinclient`, `other`, `unknown`, and `deployable`. If
this option is not specified, all types of devices are displayed.

-o, --osfilter=[operating system] - Displays devices having the specified OS installed. Valid
values are `other`, `win9x`, `winnt`, `wince`, `win2k`, `win2k3`, `winxp`, `nw6`, `nw6_5`, `nwoes`, `suse`, `sles`,
`nld`, `rh_es`, and `rh_as`. If this option is not specified, all devices are displayed.

-m, --management-status=[management status] - Displays devices having the specified status.
Valid values are `discovered`, `inventoried`, `managed`, and `retired`. If this option is not
specified, all types of devices are displayed regardless of their status.

--modefilter=[discovery mode] - Displays the devices discovered by using the specific discovery
mode. Valid values are `IP`, `LDAP`, `csvimport`, `ZENworks-migration`, and `ZAM-migration`. If
this option is not specified, all devices are displayed.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`discovery-task-list (dtl) [options]`

Lists the discovery tasks and the status.

Accepts the following options:

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards *
and ? can be used if they are enclosed in quotation marks.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`discovery-task-run-now (dtrn) (discovery task name)`

Executes the specified discovery task immediately.

Dynamic Group Commands

These commands are used to view dynamic groups, group members, and to refresh a group. Dynamic group commands begin with `dynamic-group-` in the long form, or with the letters `dg` in the short form.

dynamic-group-list (dgl) (device type) [folder] [-n│--namefilter=filter string] [-s│--scrollsize=scroll size] [-r│--recursive] [-c│--count]

Lists the dynamic groups.

Accepts the following options:

(device type) - Valid values are `server` and `workstation`.

[folder] - Lists the content of the specified folder. If it is not specified, the content of the root folder is displayed.

[-n|--namefilter=filter string] - Displays the results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.

[-s|--scrollsize=scroll size] - Number of results to be displayed at a time.

[-r|--recursive] - Lists the results recursively, including subfolders. If it is not specified, the contents of subfolders are not listed.

[-c|--count] - Displays the count of the results.

dynamic-group-members (dgm) (device type) (dynamic group name) [path of the dynamic group folder] [-s│--scrollsize=scroll size]

Lists the members of the specified dynamic group.

Accepts the following options:

(device type) - Valid values are `server` and `workstation`.

(dynamic group name) - Name of the dynamic group.

[path of the dynamic group folder] - The path of the dynamic group. You can specify either the complete path or the relative path of the root folder.

[-s|--scrollsize=scroll size] - Number of results to be displayed at a time.

dynamic-group-refresh (dgr) (device type) [(dynamic group name) (dynamic group name)... (dynamic group name)] [-f│--folder=folder name including path of the dynamic group folder] [-a│--all]

Refreshes the specified dynamic group.

Accepts the following options:

(device type) - Valid values are `server` and `workstation`.

(dynamic group name) - Name of the dynamic group. You can list multiple groups.

[-f|--folder=folder name including path of the dynamic group folder] - Dynamic group folder's name, including its full path.

[-a|--all] - Refresh all existing dynamic groups.

dynamic-group-view-query (dgvq) (path of the dynamic group folder)

Lists the filters for the specified dynamic group.

(path of the dynamic group folder) - Path of the dynamic group. You can either specify the complete path or the relative path of the root folder

Endpoint Security Management Commands

These commands are used to import and export security policies and to export a Management Zone's security policy encryption key (KMK). They apply only to ZENworks 11 SP2 Endpoint Security Management.

All of the commands below accept the option flags listed in the Global Options section, with the exception that the esmpolicy-export-to-file and esmpolicy-import commands cannot be run remotely and therefore do not accept the --host option.

esmpolicy-export-to-file (epetf) (policy path) (XML policy file path)

Exports a security policy's information to an encrypted XML policy file. The XML policy file can be used to import the policy to the same Management Zone or a different Management Zone.

NOTE: To import a policy, you must also supply the zone's security policy encryption key (KMK) so that the encrypted XML policy file can be decrypted. Use the esmpolicy-export-kmk-to-file command to create the key file.

(policy path) - The path (including the filename) of the policy object relative to the Policies root folder. For example, FWpolicy1 or ESMpolicies/DEpolicy4.

(XML policy file path) - The path (including the filename) where you want to save the XML policy file. If you specify a filename only, the file is saved to the current directory. For example, firewallpolicy.xml or c:\firewallpolicy.xml.

Examples:

zman epetf FWPolicy1 c:\FWpolicy1.xml

zman epetf ESMpolicies/DEpolicy4 DEpolicy4.xml

esmpolicy-export-kmk-to-file (epektf) (policy encryption key file path)

Exports the Management Zone's security policy encryption key (KMK) to a file. The key file is required to decrypt an XML policy file (that was exported from a policy in the zone) when importing the policy with the `esmpolicy-import` command.

(policy encryption key file path) - The path (including filename) where you want to save the security policy encryption key (KMK) file. If you specify a filename only, the file is saved to the current directory. Use any supported filename for the file. The extension is not important; you can use any extension or no extension. For example, `KMK.txt`, `key.xml`, `KMK`, and `decryption.file` are all valid filenames.

Examples:

`zman epektf c:\key.txt`

`zman epektf EncryptionKey.xml`

esmpolicy-import (epi) (policy name) (policy encryption key file path) (XML policy file path) [parent folder]

Imports a security policy from an encrypted XML file created by the `esmpolicy-export-to-file` command.

(policy name) - The name to assign to the policy object.

(policy encryption key file path) - The full path (including filename) of the security policy encryption key (KMK) file for the Management Zone from which the policy was exported. This file is required to decrypt the encrypted XML file. If the key file is in the current directory, specify the filename only.

(XML policy file path) - The full path (including filename) of the encrypted XML policy file. If the file is in the current directory, specify the filename only.

[parent folder] - The Policies folder in which to create the policy object. If you want to create the object in the root folder, ignore this option.

Examples:

`zman epi FWPolicy c:\key.txt c:\FWpolicy.xml`

`zman epi DEPolicy key.txt encryptionpolicy.xml esmpolicies/encryption`

esmpolicy-purge-effective-policies (epep) [(device path)(device path)...(device path)] [-b|--begin-date=yyyy-MM-dd HH:mm:ss] [-e|end-date=yyyy=MM-dd HH:mm:ss] [-u|--unregisteredDevices]

Purges effective policy report records from the ZENworks database. The following options can be used to target report records for specific devices or for specific time periods.

[(device path) (device path) ... (device path)]: To purge the effective policy report records for specific devices, specify the full path for each device. Ignore this option to purge reports for all devices.

[-b|--begin-date=yyyy-MM-dd HH:mm:ss]: To purge effective policy report records starting with a specific date, specify the begin date. All records with a timestamp on or after the begin date are purged. Use this option with the `end-date` option to designate a specific time period.

[-e|--end-date=yyyy=MM-dd HH:mm:ss]: To purge effective policy report records up to a specific date, specify the end date. All records with a timestamp on or before the end date are purged. Use this option with the `begin-date` option to designate a specific time period.

[-u|--unregisteredDevices]: Purge effective policy report records for devices that are no longer registered in the zone but that still have report data in the ZENworks database.

Examples:

`zman epep /Devices/Workstations/device1`

```
zman epep /Devices/Workstations/device1 -b "2010-10-10 10:10:10" -e "2010-12-31
24:00:00"

zman epep -u
```

Full Disk Encryption Commands

These commands apply only to ZENworks 11 Full Disk Encryption. In addition to the parameters listed below, the commands accept the option flags listed in the Global Options section.

`fdepolicy-purge-eri (fpe) [(device path)(device path)...(device path)] [-b|--`
`begin-date=yyyy-MM-dd HH:mm:ss] [-e|end-date=yyyy=MM-dd HH:mm:ss] [-u|--`
`unregisteredDevices]`

> Purges emergency recovery information (ERI) records from the ZENworks database. The following options can be used to target records for specific devices or for specific time periods.

> [(device path) (device path) ... (device path)]: To purge the ERI records for specific devices, specify the full path for each device. Ignore this option to purge reports for all devices.

> [-b|--begin-date=yyyy-MM-dd HH:mm:ss]: To purge ERI records starting with a specific date, specify the begin date. All records with a timestamp on or after the begin date are purged. Use this option with the `end-date` option to designate a specific time period.

> [-e|--end-date=yyyy=MM-dd HH:mm:ss]: To purge ERI records up to a specific date, specify the end date. All records with a timestamp on or before the end date are purged. Use this option with the `begin-date` option to designate a specific time period.

> [-u|--unregisteredDevices]: Purge ERI records for devices that are no longer registered in the zone but that still have report data in the ZENworks database.

> Examples:

> `zman fpe /Devices/Workstations/device1`

> `zman fpe /Devices/Workstations/device1 -b "2010-10-10 10:10:10" -e "2010-12-31`
> `24:00:00"`

> `zman fpe -u`

Hotlist Command

This command is used to view the list of devices that have unacknowledged warnings or errors. The `Hotlist` command begin with `hotlist-` in the long form, or with the letter `hl` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

`hotlist-view (hlv) [options]`

> Lists devices that have unacknowledged warnings or errors.

> Accepts the following options:

> -c, --count - Displays only the count of the hotlist items.

> -t, --type=[error type] - Filters on the error type of hotlist items. Valid values are `Noncompliant`, `Critical`, and `Warning`.

> -o, --operator=[comparison operator] - If specified, this operator is used in error-count-based filtering. Valid values are >, >=, =, <, and <=. Enclose the operators in double quotation marks so that > and < are not considered by the system as redirection operators.

-e, --errorcount=[erorr count] - Filter by the count of the error type specified. If the comparison operator is not specified, the `>=` is used by default. If error type is not specified, the count is taken as the sum of noncompliant, critical and warning errors.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

Warning or errors can be acknowledged by using the `messages-acknowledge` command. When all warnings or errors for a device are acknowledged the messages no longer appear on the hotlist.

Intel AMT Commands

These commands are used to perform Preprovisioning, Provisioning, and Management operations on the Intel AMT devices. The Intel AMT commands begin with `iamt-` in the long form, or with the letter `i` in the short form.

The supported iAMT versions in ZENworks 11 SP2 are 3, 4, 5, and 6.

`iamt-root-certificate-hash (irch)`

Prints the ZENworks Root Certificate hash.

`iamt-create-mgmtcert (icmc) [options]`

Creates or imports the Intel AMT Provisioning or Management Certificate in the `.DER` format on every Primary Server that is used as an iAMT Provisioning or Management Server.

Accepts the following options if you are importing an external certificate:

-p|--certpath - Specify the path of the Intel AMT Provisioning / Management Certificate that is signed by the External CA. The certificate should be in the `.DER` format

For example, to import an external certificate, run the following command:

```
iamt-create-mgmtcert (icmc) [-p|--certpath=Intel AMT Provisioning/Management
Certificate_path_in_.DER format]
```

`iamt-create-csr (icc) (isCA) (type) [options]`

Creates the Certificate Request for the Intel AMT Provisioning / Management Server or the AMT device in an External CA environment.

isCA - The valid values are `True` or `False`. If true, the Certificate Request generates the certificate for the Intel AMT Management Zone.

Type - Valid values are `mgmt` or `device`. The Certificate Request is created for the Management Zone or for specific devices in the Management Zone.

Accepts the following options:

-c, --country=[country name] - Specify the name of the country.

-s, --state=[state name] - Specify the name of the state.

-o, --organization=[organization name] - Specify the name of the organization.

-u, --orgunit=[Organization Unit] - Specify the organization unit.

-n, --commonname=[Common name] - Specify the common name of the certificate.

-d, --destination-folder=[Destination Folder]: Specify the path of the destination folder where you want to copy the Certificate Signing Request file.

`iamt-clear-mgmtcert (idmc)`

Clears the Intel AMT Provisioning / Management Certificate.

`iamt-cacert-import (icai) [isclear] [options]`

Imports the External CA or its subordinate CA on the server.

[isclear] - The valid values are `True` or `False`. If true, the imported External CA or its subordinate CA is deleted.

Accepts the following options if you are importing an External CA or its subordinate CA:

-i|--inform - Specifies whether the certificate format is PEM or DER.

-p|--cacertpath - Specifies the certificate path.

Imaging Command

These commands are used to perform Imaging operations. Imaging commands begin with `imaging-` in the long form, or with the letter `i` in the short form.

`imaging-apply-assigned-imaging-bundle (iaaib) (device type) (<device name> <device name> ... <device name>)`

Applies the most effective Imaging bundle.

Accepts the following options:

(Device type) - Valid values are `server` and `workstation`.

(Device name) - Name of the server or workstation object.

For example, to apply the assigned Imaging bundle to a server with the name server1, run the following command:

`zman iaaib server server1`

If the device object is located within a folder, use the following the command:

`iaaib (Device type) (folder/path of device)/(Device name)`

where `folder/path of device` is the relative path of the device within the `Workstation` or `Server` folder.

If you have multiple workstations in a directory, and the name of the workstations is prepended by common alphanumeric characters (for example, dev091, dev092,...dev099), use the following command from a Linux server to simultaneously apply the assigned imaging bundle on multiple devices:

`zman iaaib workstation folderx/dev{09{1,2,3,4,5,6,7,9}}`

The bundle is applied to devices dev091, dev092, and so on.

Inventory Commands

These commands are used to initiate scans and send an inventory data collection form to the managed devices. Inventory commands begin with `inventory-` in the long form, or with the letter `i` in the short form.

inventory-collection-wizard-run-now (icwrn) (device object path) [...] [options]

Sends the inventory data collection form to one or more devices.

Accepts the following option:

-f, --folder=[device folder] - The path of the device folder relative to `/Devices`. If this option is specified, the path of the device objects is determined relative to this folder. This option can be used to specify multiple device objects under the same folder.

inventory-scan-now (isn) [options] (device object path) [...]

Initiates an inventory scan of one or more devices.

Accepts the following option:

-f, --folder=[device folder] - The path of the device folder relative to `/Devices`. If this option is specified, the path of the device objects is determined relative to this folder. This option can be used to specify multiple device objects under the same folder.

inventory-import-localproducts (iilp) (pathname) [-U (Administrator)] [-P (password)]

Imports the administrator-defined local products and product fingerprints. Only administrators with CDLP rights can import the local products.

To import product data:

`zman iilp {pathname} [-U (Administrator)] [-P (password)]`.

Example:

To import administrator-defined local products to ZENworks Configuration Management from a file in `./output/lpimports.txt`:

`zman iilp ./output/lpimports.txt -U Administrator -P novell`.

inventory-export-localproducts (ielp) (pathname) [-U (Administrator)] [-P (password)]

Exports the administrator-defined local products and product fingerprints. Only administrators with CDLP rights can export the local products.

To export product data:

`zman ielp {pathname} [-U (Administrator)] [-P (password)]`.

Example:

To export all administrator-defined local products from ZENworks Configuration Management to a file in `./output/lpexports.txt`:

`zman ielp ./output/lpexports.txt -U Administrator -P novell`.

License Commands

These commands are used to activate your ZENworks server or to display licensing information. License commands begin with `license-` in the long form, or with the letter `l` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

license-activate (la) (product name) (license key)

Activates the ZENworks products.

license-deactivate (ld) (product name)

Deactivates the ZENworks products. Only the product name is needed as an argument.

license-info (li)

Displays the licensing information.

license-refresh (lr)

Forces the system to refresh the stored license cache, which is normally updated once per day. Use this command to update the cache at any time, such as when an evaluation license might expire and the cache doesn't yet reflect that.

license-show-active-components (lsac)

Lists the current license states of all known ZENworks Configuration Management components and DataModel plug-ins (two separate lists).

Location Commands

These commands are used to view and manage locations. A location consists of one or more network environments. When the ZENworks Adaptive Agent determines that its current environment matches a network environment associated with a location, the agent enforces any security policies and closest server rules assigned to the location.

Location commands begin with `location-` in the long form, or with the letter `l` in the short form. All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

location-create (lc) (location name) [-d|--desc=<description>] [network environment name] [...]

Creates a location and (optionally) assigns network environments to the location.

(location name) - The name you want assigned to the new location.

-d, --desc=<description> - A description for the location.

[network environment name] - The name of an existing network environment that you want to assign to the location. You can add multiple network environments. Adding network environments during creation of a location is optional. You can use the location-nwenv-assign command to add network environments after creation.

location-delete (ldel) (location name) [...]

Deletes locations.

(location name) - The name of the location you want to delete. You can specify one or more location names.

location-delete-all-network-assignments (ldana) (location name)

Removes all network environments from a location.

(location name) - The name of the location from which you want to remove all assigned network environments.

location-delete-network-assignments (ldna) (location name) [network environment name] [...]

Removes specific network environments from a location.

(location name) - The name of the location from which you want to remove network environments.

[network environment name] - The name of the network environment that you want to remove. You can remove multiple network environments.

location-list (ll) [options]

Lists locations.

Accepts the following options:

-c, --count - Displays the count of the results.
-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

location-list-network-assignments (llna) (location name)

Lists the network environments assigned to a location.

(location name) - The name of the location whose network environments you want to view.

Accepts the following options:

-c, --count - Displays the count of the results.
-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`location-list-relationships (llrel) (location name) [-c count]`

Lists the ZENworks objects such as policies and bundles that are associated with the specified location.

(location name) - The name of the location whose ZENworks objects you want to view.

Accepts the following option:

-c, --count - Displays the count of the results.

`location-network-assign (lna) (location name) (network environment name) [...]`

Adds network environments to a location.

(location name) - The name of the location to which you want to assign network environments.

[network environment name] - The name of an existing network environment that you want to add to the location. You can add multiple network environments.

`location-rename (lrn) (old name) (new name)`

Renames a location.

(old name) - The name of the location that you want to rename.

(new name) - The new name for the location.

`location-throttle (lt) (location name) (throttle rate)`

Sets the bandwidth throttle rate to be used for distributing content to devices identified with the specified location. To maximize performance of your ZENworks Servers and network system, high bandwidth environments can use one set of throttle rates and low bandwidth environments can use a different set of throttle rates. The throttle rate can be overridden in a bundle so that high-priority patch and bundle content can be deployed quickly.

(location name) - The name of the location for which you want to define the bandwidth throttle rate.

(throttle rate) - The maximum bandwidth rate in kilobits per second (kbps).

`location-unknown-throttle (lut) (throttle rate)`

Sets the bandwidth throttle rate to be used for distributing content to devices identified with the Unknown location.

(throttle rate) - The maximum bandwidth rate in kilobits per second (kbps).

Location Rules Commands

These commands are used to copy, export, and import Closest Server rules. Location commands begin with `location-` in the long form, or with the letter `loc` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

The Location Rules commands are applicable only for the ZENworks Configuration Management 10.2.x/10.3.x zone. The commands are no longer valid after you baseline your zone to ZENworks 11 SP2. To configure locations in ZENworks 11 SP2, you must use the commands listed in the Location category.

location-copy-rules (loccp) [options] (source device or device folder path) (destination device or device folder path)

Imports Closest Server rules data (in XML format) from a file. If the devices or folders specified in the XML file no longer exist, they are skipped.

(source device or device folder path) - Path of the device or device folder starting with `/Devices` from which Closest Server rules are to be copied.

(destination device or device folder path) - Paths of one or more devices or device folders starting with `/Devices` to which Closest Server rules are to be applied. If the destination does not exist, or does not appear in the XML file, it will be skipped.

Accepts the following options:

-q, --quitOnError - Stop processing if an error occurs while saving any rule. The default is to note the error and continue with the next rule.

-o, --overwriteExisting - Overwrite existing rules on devices and folders that are specified in the XML file. The default is to skip destinations with existing rules.

Examples:

To copy the rules from `/Devices/Workstations/Folder1` to `/Devices/Workstations/Folder2` and `/Devices/Workstations/MyComputer`:

`zman loccp /Devices/Workstations/Folder1 /Devices/Workstations/Folder2 /Devices/Workstations/MyComputer`

To copy the rules from `/Devices/Workstations/Folder1` to `/Devices/Workstations/Folder2` overwriting existing values:

`zman loccp -o /Devices/Workstations/Folder1 /Devices/Workstations/Folder2`

location-export-rules-to-file (loctf) [options] (xml file path)

Exports Closest Server rules data (in XML format) to a file. The XML file can be used as input for creating or appending to the Closest Server rules.

(XML file path) - The file into which the rules data will be stored in XML format. The file is created if it does not exist.

Accepts the following option:

-s, --source - Path of the device or device folder starting with `/Devices` from which Closest Server rules are to be exported. If not specified, all Closest Server rules will be exported.

Examples:

To export the Closest Server rules for folder `/Devices/Servers` to `C:\rules.xml`:

`zman loctf -s /Devices/Servers C:\rules.xml`

To export all Closest Server rules to `C:\rules.xml`:

```
zman loctf C:\rules.xml
```

location-import-rules-from-file (locff) [options] (XML file path) [<destination device or device folder path> <destination device or device folder path> <destination device or device folder path>]

Imports Closest Server rules data (in XML format) from a file. If the devices or folders specified in the XML file no longer exist, they are skipped.

(XML file path) - XML file containing exported rules information. Use `location-export-rules-to-file (loctf)` to export settings information into an XML file.

[destination device or device folder path][...] - Paths of one or more devices or device folders starting with /Devices to which Closest Server rules are to be applied. If the destination does not exist, or does not appear in the XML file, it will be skipped.

Accepts the following options:

-q, --quitOnError - Stop processing if an error occurs while saving any rule. The default is to note the error and continue with the next rule.

-o, --overwriteExisting - Overwrite existing rules on devices and folders that are specified in the XML file. The default is to skip destinations with existing rules.

Examples:

To import Closest Server rules from `C:\rules.xml`:

```
zman locff C:\rules.xml
```

To import Closest Server rules from `C:\rules.xml` and quit on error:

```
zman locff -q C:\rules.xml
```

To import Closest Server rules from `C:\rules.xml` overwriting existing values:

```
zman locff -o C:\rules.xml
```

Message Commands

These commands are used to view and manage messages. Message commands begin with `message-` in the long form, or with the letter `m` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

- "messages-acknowledge (ma) [options] [object path] [log id] [...]" on page 44
- "messages-delete (md) [options] [object path]" on page 45
- "messages-view (mv) [options] [object path]" on page 45

messages-acknowledge (ma) [options] [object path] [log id] [...]

Acknowledges messages associated with a ZENworks managed device, bundle, or policy.

[object path] - Full path of the object that has messages associated with it. The object can be a server, workstation, bundle, or policy.

Accepts the following options:

-b, --begin-date=[message logged date] - Messages logged on and after this date are acknowledged. The format for the date is YYYY-MM-DD.

-e, --end-date=[message logged date] - Messages logged on and before this date are acknowledged. The format for the date is YYYY-MM-DD.

-a, --all - Acknowledges all messages logged for the object. If an object is not specified, all the logged messages are acknowledged.

-l, --logID=[log id][...] - A comma-separated list of log IDs of the messages to be acknowledged. You can obtain the log IDs by using the `messages-view` command.

-L, --logID-file=[file path] - The path of the file containing message log IDs. Each line in the file should contain a message log ID. Use this option instead of the `--log-ID` option to input the log IDs.

`messages-delete (md) [options] [object path]`

Deletes messages associated with a ZENworks managed device.

[object path] - Full path of the object that has messages associated with it. The object can be a server, workstation, bundle, or policy.

Accepts the following options:

-b, --begin-date=[message logged date] - Messages logged on and after this date are deleted. The format for the date is YYYY-MM-DD.

-e, --end-date=[message logged date] - Messages logged on and before this date are deleted. The format for the date is YYYY-MM-DD.

-l, --logID=[log ID][...] - A comma-separated list of log IDs of the messages to be deleted. You can obtain the log IDs by using the `messages-view` command.

-L, --logID-file=[file path] - The path of the file containing the message log IDs. Each line in the file should contain a message log ID. Use this option instead of the `--log-ID` option to input the log IDs.

`messages-view (mv) [options] [object path]`

Lists messages associated with a ZENworks managed device, bundle, or policy.

-S, --severity=[severity [...] - Filters on the message severity. Valid values are `critical`, `warning`, and `info`.

-t, --type=[message type] - Filters on the source of message. Valid values are `server` and `client`. Filtering on `server` lists messages generated by the ZENworks server and filtering on `client` lists messages generated by the ZENworks Adaptive Agent on that device.

-D, --date-condition=[date condition] - Filters messages based on a date range. Valid values are `before`, `since`, and `between`. Use the --begin-date and --end-date options to specify the dates. If date condition is used and no date is specified, the current date is taken by default.

-b, --begin-date=[message logged date] - The date on which the message was logged. The format for the date is YYYY-MM-DD. This option is to be used in conjunction with the date condition option. If date condition is not specified, messages logged before this date are displayed.

-e, --end-date=[message logged date] - The date on which the message was logged. The format for the date is YYYY-MM-DD. This option is to be specified when `between` is specified as the value for the date condition option.

-a, --acknowledged - Filters on acknowledged messages.

-n, --not-acknowledged - Filters on messages that are not acknowledged.

-m, --messagefilter=[localized message] - Filters on the localized message.

-c, --count - Displays only the count of the messages.

-o, --sort-order=[sort order] - Sorts the messages based on date, severity, and acknowledged.

-A, --asc - Specifies the direction of sorted listing. The default is descending order.

-d, --detailed - Returns more information about the messages.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

Network Environment Commands

These commands are used to view and manage network environments. A network environment consists of services (gateways, DNS servers, DHCP servers, and so forth) that identify the environment.

Network environments are the building blocks for locations. When the ZENworks Adaptive Agent determines that its current environment matches a network environment assigned to a location, the agent enforces any security policies and closest server rules assigned to the location.

Network Environment commands begin with `network-environment-` in the long form, or with the letters `ne` in the short form. All of the commands below accept the flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

- "network-environment-create (nec) (network environment name) [-d|--desc=<description>] [network environment XML file]" on page 46
- "network-environment-delete (ned) (network environment name) [...]" on page 46
- "network-environment-export-to-file (neetf) (network environment name) (XML file path)" on page 46
- "network-environment-list (nel) [options]" on page 47
- "network-environment-list-relationships (nelr) (network environment name) [-cr count]" on page 47
- "network-environment-rename (ner) (old name) (new name)" on page 47
- "network-environment-throttle (net) (network environment name) (throttle rate)" on page 47

network-environment-create (nec) (network environment name) [-d|-- desc=<description>] [network environment XML file]

Creates a network environment from the information contained in the specified XML file.

(network environment name) - The name you want assigned to the new network environment.

-d, --desc=<description> - A description for the network environment.

[network environment XML file] - The pathname of the XML file containing the information needed to create the network environment. If you don't already have an XML file, use the `network-environment-export-to-file` command to export the information for an existing network environment. You can then use the XML file as a template for creating new network environment files.

network-environment-delete (ned) (network environment name) [...]

Deletes network environments.

(network environment name) - The name of the network environment you want to delete. You can specify one or more names.

network-environment-export-to-file (neetf) (network environment name) (XML file path)

Exports a network environment's information to an XML-formatted file.

The `network-environment-create` command requires network environment information to be input through an XML-formatted file. You can use this file as a template, modifying it as necessary to include the information you want used when creating a new network environment.

(network environment name] - The name of the network environment whose information you want to export to the file.

(XML file path) - The file path and name for the export file.

`network-environment-list (nel) [options]`

Lists network environments.

Accepts the following options:

-c, --count - Displays the count of the results.
-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`network-environment-list-relationships (nelr) (network environment name) [-cr count]`

Lists the ZENworks objects such as policies and bundles that are associated with the specified network environment.

(network environment name) - The name of the network environment whose ZENworks objects you want to view.

Accepts the following option:

-c, --count - Displays the count of the results.

`network-environment-rename (ner) (old name) (new name)`

Renames a network environment.

(old name) - The name of the network environment that you want to rename.

(new name) - The new name for the network environment.

`network-environment-throttle (net) (network environment name) (throttle rate)`

Sets the bandwidth throttle rate for distributing content to devices located in a specific network environment. To maximize performance of your ZENworks Servers and network system, high bandwidth environments can use one set of throttle rates and low bandwidth environments can use a different set of throttle rates. The throttle rate can be overridden in a bundle so that high-priority patch and bundle content can be deployed quickly.

(network environment name) - The name of the network environment for which you want to define the bandwidth throttle rate.

(throttle rate) - The maximum bandwidth rate in kilobits per second (kbps).

Novell Customer Center Registration Command

This command is used to register the Primary Server to the Novell Customer Center.

The Novell Customer Center Registration command begins with `ncc-register` in the long form, or with `nccreg` in the short form. This command accepts the flags listed in the Global Options section. In addition, it accepts individual options as listed with the command.

-

`(ncc-register|nccreg) (abc@xyz.com) (registration-key) [Primary Server path] [-i|--nccConfigXML=ncc-config.xml] [-r|--reRegister]`

Registers the Primary Server to the Novell Customer Center.

(abc@xyz.com) - The Email address to register the Primary Server to the Novell Customer Center Server.

(registration-key) - The Registration key to register the Primary Server to the Novell Customer Center Server.

[Primary Server path] - The path of the Primary Server to register to the Novell Customer Center Server.

-i, --nccConfigXML= ncc-config.xml - The XML file that contains registration information such as the product, version, architecture, and release.

-r, --reRegister - Reregister the Primary Server to the Novell Customer Center Server.

Examples:

To register the Local Primary Server to the Novell Customer Center Server, run the following command:

```
zman nccreg email reg-key
```

To register a specific Primary Server to the Novell Customer Center Server, run the following command:

```
zman nccreg email reg-key server_path
```

To reregister the Local Primary Server to the Novell Customer Center Server, run the following command:

```
zman nccreg email reg-key -r
```

To reregister a specific Primary Server to the Novell Customer Center Server, run the following command:

```
zman nccreg email reg-key server_path -r
```

Object Commands

These are utility or convenience commands to operate on ZENworks objects. Object commands begin with object- in the long form, or with the letter o in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

- "object-get-GUID (ogg) (object path)" on page 48
- "object-get-name (ogn) (ZENworks GUID)" on page 48
- "object-get-path (ogp) (ZENworks GUID)" on page 48

object-get-GUID (ogg) (object path)

Gets the GUID of a ZENworks object.

(object path) - Full path of the ZENworks object, starting with a forward slash (/).

object-get-name (ogn) (ZENworks GUID)

Gets the name of a ZENworks object.

(ZENworks GUID) - The ZENworks GUID of the object, consisting of 32 characters.

object-get-path (ogp) (ZENworks GUID)

Gets the path of a ZENworks object, given the GUID.

(ZENworks GUID) - The ZENworks GUID of the object, consisting of 32 characters.

Policy Commands

These commands are used to create, modify and manage policies. Policy commands begin with `policy-` in the long form, or with the letter `p` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

policy-assign (pa) (device or user type) (policy or policy group) (device or user object path) [...] [options]

Assigns a policy or policy group to one or more device or user objects.

(device or user type) - Valid values are `device`, `server`, `workstation`, and `user`.

(device or user object path) [...] - The path of the device or user objects relative to the root folder of the device or user type specified.

Accepts the following options:

-c, --conflicts=[policy conflict resolution order] - Determine how policy conflicts are resolved. Valid values are `userlast` or `1`, `devicelast` or `2`, `deviceonly` or `3`, `useronly` or `4`. For `userlast`, device-associated policies are applied first followed by user-associated policies. For `devicelast`, user-associated policies are applied first followed by device-associated policies. For `deviceonly`, user-associated policies are ignored. For `useronly`, device-associated policies are ignored. If this option is not specified, `userlast` is taken as the default value.

-e, --enforce-now - Enforces the policy immediately on all assigned devices.

-f, --folder=[device or user folder] - The path of the device or user folder relative to the respective root folder. If this option is specified, the path of the device or user objects is determined relative to this folder. This can be used to specify multiple device or user objects under the same folder.

policy-create (pc) (policy name) (policy XML file) [parent folder] [options]

Creates a policy.

(policy XML file) - XML file containing exported policy information. Use `policy-export-to-file` (petf) to export a policy's information into a XML file. If you want to reuse files, template XML files can be created from policies created through ZENworks Control Center.

Accepts the following options:

-d, --desc=[description] - Description for the policy.

-a, --actioninfo=[file content for policy] - XML file containing information about file content to be associated and packaged with the policy. For example, the driver file to be installed for a Printer policy is the file content. For the XML format template, refer to `ActionInfo.xml` located in `/opt/novell/zenworks/share/zman/samples/policies` on a Linux server or *Installation_directory*`:\Novell\Zenworks\share\zman\samples\policies` on a Windows server.

-s, --create-as-sandbox=[create as sandbox] - Creates the policy as a sandbox.

policy-copy (pcp) (policy path) (policy name) [parent folder] [options]

Copies a policy.

Accepts the following option:

-d, --desc=[description] - Description for the policy.

policy-delete (pd) (policy object path) [...] [options]

Deletes one or more policy objects.

(policy object path) [...] - The path of the policy objects (policy, policy folder or policy group) relative to `/Policies`. The wildcard * can be used in the object names if it is enclosed in quotations. Exercise caution while using wildcards for deleting objects.

Accepts the following options:

-r, --recursive - Deletes objects inside a folder recursively.

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

`policy-disable (pdl) (policy path) [...] [options]`

Disables policies.

Accepts the following option:

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

`policy-delete-version (pdv) (policy path) [version of the policy to be deleted] [options]`

Deletes one or more versions of the policy.

Examples:

To delete the version 3 of the policy, zenpolicy:

`zman pdv zenpolicy 3`

To delete all the versions of the policy, zenpolicy:

`zman pdv zenpolicy -a`

To delete all the versions of the policy, zenpolicy, that are older than the specified version:

`zman pdv zenpolicy version -p`

`policy-enable (pel) (policy path) [...] [options]`

Enables policies.

Accepts the following option:

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

`policy-export-to-file (petf) (policy path) (XML file path) [version of the policy] [-c|--export-content]`

Exports a policy's information (in XML format) to a file. The XML file is to be used as input for creating policies.

(XML file path) - The complete path of the XML file to which the policy information is to be exported.

[Version of the policy] - Version of the policy to be exported. If the version is not specified, the published version of the policy is exported. To export a sandbox version of the policy, specify sandbox.

[-c|--export-content] - Exports the content of the policy to a subdirectory within the directory containing the XML file to which the policy's information is exported. If the option is not specified, the policy content is not exported.

Examples:

To export the sandbox version of a policy named zenpolicy to an XML file named `policy.xml`:

`zman petf policyfolder/zenpolicy C:\policies\policy.xml sandbox`

To export version 3 of a policy named zenpolicy to an XML file named `policy.xml`:

`zman petf policyfolder/zenpolicy C:\policies\policy.xml 3`

To export version 5 of a policy named zenpolicy to an XML file named `policy.xml` and export the policy's content to the `C:\policies\policy_content directory`:

`zman petf policyfolder/zenpolicy C:\policies\policy.xml 5 -c`

policy-folder-create (pfc) (folder name) [parent folder] [options]

Creates a new folder for containing policies.

Accepts the following option:

--desc=[description] - Description for the folder.

policy-group-add (pga) (policy group path) (policy path) [...] [options]

Adds policies to a policy group.

Accepts the following option:

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

policy-group-create (pgc) (group name) [parent folder] [options]

Creates a policy group and adds members to it.

Accepts the following options:

--desc=[description] - Description for the group.
-m, --members=[policy path][...] - The path of the policies relative to /Policies.
-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

policy-group-members (pgm) (policy group path) [options]

Lists members of a policy group.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

policy-group-remove (pgr) (policy group path) (policy path) [...] [options]

Removes policies from a policy group.

Accepts the following option:

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

policy-list (pl) [options] [folder]

Lists policy objects.

Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, the contents of the subfolders are not listed.
-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.
-c, --count - Displays the count of the results.
-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

policy-list-assignments (plas) (policy or policy group) [options]

Lists the device and user assignments for a policy.

Accepts the following options:

-t, --typefilter=[assignment type] - Filters on the assignment type. Valid values are `device` and `user`.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

policy-list-groups (plg) (policy path) [options]

Lists groups where the given policy is a member.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

policy-list-version (plv) (policy path)

Lists all the versions of the policy.

Examples:

To list all the versions of the policy, zenpolicy:

```
zman plv zenpolicy
```

policy-move (pmv) (policy object path) [destination folder path]

Moves a policy object to a different folder.

policy-sandbox-create (psc) (policy path) [version of the policy] [options]

Create a sandbox from an existing version of a policy

[version of the policy] - Specify the version of the policy to be used for creating the sandbox.

Accepts the following option:

-f, --force =[force] - Overwrite any existing sandbox.

Example:

To create a sandbox from the version 2 of a policy, zenpolicy:

```
zman psc zenpolicy 2
```

policy-sandbox-publish (psp) (policy path)

Publish a sandbox to create a published version of the policy.

Example:

To publish the sandbox of a policy, zenpolicy

```
zman psp zenpolicy
```

policy-sandbox-publish-to-newpolicy (psptn) (policy path) (policy name) [parent folder] [options]

Publishes sandbox version of a policy to a new policy.

(policy path) - The path of the policy object relative to the policies root folder, `/Policies`.

(policy name) - Name of the new policy to be created.

[parent folder] - Folder in which the new policy is to be created. If not specified, the policy is created in the root folder.

Accepts the following options:

-s, --createAsSandbox =[create as sandbox] - Creates the policy as a sandbox.

-g, --groups - Adds the newly created policy to all the groups which the source policy is a member of.

-d, --deviceAssignments - Copies all the device assignments from the source policy to the new policy created.

-u, --userAssignments - Copies all the user assignments from the source policy to the new policy created.

Example:

To publish the sandbox version of a policy to a new policy.

```
zman psptn policy1 policy2 /Policies/Folder1
```

policy-sandbox-revert (psr) (policy path)

Deletes the sandbox and reverts to the latest published version of the policy.

Example:

To delete the sandbox and revert to the latest published version of the policy:

```
zman psr zenpolicy
```

policy-rename (pr) (policy object path) (new name) [options]

Renames a policy object.

-p, --publish =[publish] - If this option is specified and the display name of the bundle is edited, the bundle is immediately published. However, if the current display name of the bundle is different from the existing bundle name or if a sandbox already exists for the bundle, specifying this option will have no effect.

policy-settings-copy (pscp) (policy source path) (destination policy or policy folder path) [...] [options]

Copies settings.

(source policy or policy folder path) - The path of the policy or policy folder relative to / Policies from which settings have to be copied.

(destination policy or policy folder path) [...] - The path of the policies or policy folders relative to /Policies to which settings must be copied.

Accepts the following option:

-n, --names=[settings name][...] - Names of the settings to be copied from the source path. If this option is not specified, all settings defined at the source path are copied.

policy-settings-export-to-file (psetf) [options] (XML file path) [settings name] [...]

Exports settings data (in XML format) to a file. The XML file is to be used as input for creating or modifying settings.

(XML file path) - The file into which the settings data is stored in XML format. If the file does not exist, a new file is created.

[settings name] [...] - Names of the settings to be exported. If this option is not specified, all settings are exported.

Accepts the following options:

-s, --source=[source policy or policy folder path] - The path of the policy or policy folder relative to /Policies from which settings must be exported. If this option is not specified, settings are exported from the Zone.

-e, --effective - If specified, the effective settings are retrieved; otherwise, only the settings defined or overridden at the source path are retrieved.

policy-settings-revert (psrt) (policy source path) (settings name) [...]

Reverts the settings to that of the parent folder.

(source policy or policy folder path) - The path of the policy or policy folder relative to `/Policies`, whose settings must be reverted.

policy-settings-set (pss) [options] (XML file path) [destination policy or policy folder path] [...]

Sets settings at various levels.

(XML file path) - XML file containing exported settings information. Use `settings-export-to-file` (`setf`) to export settings information into a XML file.

Accepts the following options:

[settings name] [...] - Names of the settings to be exported. If this option is not specified, all settings are exported.

-f, --force - Forces all children (subfolders and individual polices) to inherit these settings.

-s, --source=[source policy or policy folder path] - The path of the policy or policy folder relative to `/Policies` for which settings have to be set. If this option is not specified, settings are set at the Zone level.

policy-unassign (pua) (device or user type) (policy or policy group) (device or user object path) [...] [options]

Unassigns a policy or policy group from one or more device or user objects.

(device or user type) - Valid values are `device`, `server`, `workstation`, and `user`.

(device or user object path) [...] - The path of the device or user objects relative to the root folder of the device or user type specified.

Accepts the following option:

-f, --folder=[device or user folder] - The path of the device or user folder relative to the respective root folder. If this option is specified, the path of the device or user objects is determined relative to this folder. This can be used to specify multiple device or user objects under the same folder.

policy-view-advanced-deployment-status (pvads) (policy path) [options]

Displays the advanced deployment status of a policy.

Accepts the following options:

-d, --device=[device path] - Displays the deployment status only for the specified device. The path of the device is relative to `/Devices`.

-u, --user=[user path] - Displays the deployment status for the specified user only. The path of the user is relative to `/Users`. If device is also specified, status details for the specified user logged into the specified device are displayed.

-n, --namefilter=[target device name] - Filters on the name of the device. Displays options matching the specified filter. The wildcard * and ? can be used if they are enclosed in quotation marks.

--statusfilter=[status type][...] - Filters on the status of Policy Apply Event. Valid values are S, F, and P (Success, Failure and Pending). A comma-separated list of status types can be given.

-t, --typefilter=[target device or user type][...] - Filters on the type of the target. Valid values are `server`, `workstation`, and `user`. A comma-separated list of target types can be given.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

```
policy-view-summary-totals (pvst) (policy path)
```
Displays a summary of the deployment status of a policy.

Query Commands

These commands are used to list and run saved searches. Query commands begin with `query-` in the long form, or with the letter `q` in the short form.

- "query-list-savedsearches|qls [-d|--detailed]" on page 56
- "query-run-savedsearch|qrs (saved search) [folder path] [-r|--recursive] [-s|--scrollsize=scroll size]" on page 56

```
query-list-savedsearches|qls [-d|--detailed]
```
Lists the saved searches.

Accepts the following option:

[-d, --detailed] - Lists the filter criteria for the saved search.

```
query-run-savedsearch|qrs (saved search) [folder path] [-r|--recursive] [-s|--
scrollsize=scroll size]
```
Runs a saved search.

Accepts the following options:

(saved search) - Saved search to be executed.
[folder path] - The path of the device folder where the search is to be performed relative to `/Devices`. The default folder is `/Devices`.
[-r|--recursive] - Lists the results recursively, including subfolders. If this option is not specified, the contents of subfolders are not listed.
[-s|--scrollsize=scroll size] - Number of results to be displayed at a time.

Queue Commands

These commands are used when you need to make modifications to the queue. The queue processes asynchronous events such as client refreshes and content cleanup, and does not need to be modified under most circumstances. Queue commands begin with `queue-` in the long form, or with the letter `q` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

- "queue-flush (qf) (queue status)" on page 56
- "queue-list (ql) [server path] [options]" on page 56
- "queue-reset (qr) [server path]" on page 57

```
queue-flush (qf) (queue status)
```
Flushes the queue by removing the queue entries.

(queue status) [...] - Filters on the status of the queue entries. Valid values are N, I, C, S, and F (New, In Progress, Cancelled, Success, and Failed).

```
queue-list (ql) [server path] [options]
```
Lists queue entries.

[server path] - The path of a server relative to /Devices/Servers. Lists only the queue entries belonging to the specified server.

Accepts the following options:

-t, --type=[queue action type] - Filters on the type of the queue entry. For example, content.cleanup is a queue entry type.

-s, --status=[queue status] - Filters on the status of the queue entries. Valid values are N, I, C, S, and F (New, In Progress, Cancelled, Success, and Failed).

queue-reset (qr) [server path]

Resets the queue, and sets the status of failed queue entries to New.

[server path] - The path of a server relative to /Devices/Servers. Resets only the queue entries belonging to the specified server.

Registration Commands

These commands allow you to create and alter registrations. They require a device type to be specified. It determines the type of device folder and device group the registration key is associated with and therefore the type of the registering devices. Registration commands begin with registration- in the long form, or with the letter r in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

registration-add-device-groups (radg) (device type) (registration key) (device group path) [...] [options]

Adds membership in the specified device groups for devices registering with the given key.

(device type) - Type of the registering device. Valid values are server and workstation.

Accepts the following option:

-f, --folder=[device folder] - The path of the device folder relative to root folder of the device type specified. If this option is specified, path of the device objects is determined relative to this folder. This can be used to specify multiple device objects under the same folder.

registration-create-key (rck) (device type) (registration key) [registration folder path] [options]

> Creates a new registration key.
>
> (device type) - Type of the registering device. Valid values are `server` and `workstation`.
>
> Accepts the following options:
>
> > -f, --devicefolder=[device folder] - The path of the device folder relative to the root folder of the device type specified. Registering devices are placed in this folder.
> >
> > -g, --devicegroup=[device group path][...] - A comma-separated list of path of the device groups. The paths specified should be relative to the root folder of the device type specified. Registering devices become members of these device groups.
> >
> > --desc=[description] - Description for the registration key.
> >
> > --site=[site] - The site where the devices are located. Registering devices are populated with this site information.
> >
> > --dept=[department] - The department in which the devices are used. Registering devices are populated with this department information.
> >
> > --loc=[location] - The physical location of the devices. Registering devices are populated with this location information.
> >
> > --limit=[limit] - Number of times this key can be used to register devices.

registration-copy (rcp) (registration key) (new name)

> Copies a registration key.
>
> (registration key) - Name of the registration key.
>
> (new name) - Name for the copied registration key.

registration-delete (rd) (registration object path) [...] [options]

> Deletes registration objects.
>
> (registration object path) [...] - The path of the registration objects (registration key or folder) relative to /Keys. The wildcard * can be used in the object names if it is enclosed in quotations. Exercise caution while using wildcards for deleting objects.
>
> Accepts the following options:
>
> > -r, --recursive - Deletes objects inside a folder recursively.
> >
> > -f, --folder=[registration folder] - The path of a registration folder relative to /Keys. If this option is specified, the path of the registration objects is determined relative to this folder. This can be used to specify multiple registration objects under the same folder.

registration-folder-create (rfc) (folder name) [parent folder] [options]

> Creates a new registration folder.
>
> Accepts the following option:
>
> > --desc=[description] - Description for the folder.

registration-info (ri) (registratin key)

> Displays detailed information about a registration key.

registration-list (rl) [options] [folder]

> Lists all registration objects.
>
> Accepts the following options:
>
> > -r, --recursive - Lists results recursively, including subfolders. If this option is not specified, the contents of the subfolders are not listed.

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards *
and ? can be used if they are enclosed in quotation marks.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

registration-list-device-groups (rldg) (registration key)(rd)

Lists the device groups associated with a registration key.

registration-move (rmv) (registration object path) [destination folder path]

Moves a registration object to a different registration folder.

registration-remove-device-groups (rrdg) (device type) (registration key) (device group path) [...] [options]

Removes the association of device groups to a registration key.

(device type) - Type of the registering device. Valid values are `server` and `workstation`.

Accepts the following option:

-f, --folder=[device folder] - The path of the device folder relative to the root folder of the device type specified. If this option is specified, the path of the device objects is determined relative to this folder. This can be used to specify multiple device objects under the same folder.

registration-rename (rr) (registration object path) (new name)

Renames a registration object.

registration-update-key (ruk) (device type) (registration key) [options]

Updates a registration key.

Accepts the following options:

-f, --devicefolder=[device folder] - The path of the device folder relative to the root folder of the device type specified. Registering devices are placed in this folder.

--desc=[description] - Description for the registration key.

--site=[site] - The site where the devices are located. Registering devices are populated with this site information.

--dept=[department] - The department in which the devices are used. Registering devices are populated with this department information.

--loc=[location] - The physical location of the devices. Registering devices are populated with this location information.

--limit=[limit] - Number of times this key can be used to register devices.

--unlimited - The key can be used without limit to register devices.

registration-import-devices (rid) (CSV file path) (registration key)

Imports the devices from the CSV file, which is used to specify the device information.

(CSV file path) - The name of the CSV file with the path. Each entry in the file contains details for a workstation or a server.

For a workstation, or a server, the format of the file entry is version, host name, serial number, and macAddress.

For a workstation, or a server, the version is WS_1.0.

The values for version and host name are mandatory, and the values for serial number and macAddress are optional.

(registration key) - Name of the registration key.

NOTE: Ensure that the file is in UTF-8 format.

For example:

To register the devices in the CSV `devicesInfo` file by using the registration key `regkey`, run the `zman rid c:\devicesInfo.csv regkey` command.

Report Commands

These commands are used to work with reports. Report commands begin with `report-` in the long form, or with the letters `rp` in the short form.

NOTE: Report commands are applicable only to the reports that are in the Custom Reports and Predefined Reports folders. Report commands do not have any effect on the reports that are in the Favorites folder.

---

report-add-user (rpau) (Username)

Adds a new user account to the ZENworks Reporting Server.

(Username) - The name of the user to be added to the ZENworks Reporting Server.

report-copy (rpc) (report name) (report folder) (report type) [options]

Copies a report.

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

Accepts the following options:

-n, --new-name=[new name] - New name for the copied report.

-d, --destination-folder=[destination folder] - The folder to which the report should be copied. If this option is not specified, a copy of the report is created in the same folder.

report-custom-folder-create (rpcfc) (Folder Name)

Creates a new subfolder within the custom-reports folder. If you do not specify the subfolder name or the path of the folder and the custom-reports folder does not exist, only the custom-reports folder is created.

(Folder Name) - Name of the folder or the complete path of the folder to be created.

Usage Example: To create a new subfolder with the name c in the customer-folders/a/b folder, enter zman rpcfc a/b/c. If either of the a or b folders do not exist, the command also creates those folders in customer-folders according to the specified path.

report-delete-definition (rpdd) (report name) (report folder) (report type)

Deletes the report definition when the report type and location are specified.

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

report-delete-definition-by-id (rpddbi) (report ID)

Deletes the report definition when the report ID is specified.

report-delete-from-history (rpdfh) (report name) (report folder) (report type) [options]

Deletes the report instances from the history when the report type and location are specified

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

Accepts the following options:

-s, --start-time=[start time] - Filters the report instances generated after a given time. Specify the time in the format of YYYY-MM-DD HH:MM enclosed in double quotation marks.

-e, --end-time=[end time] - Filter the report instances generated before a given time. Specify the time in the format of YYYY-MM-DD HH:MM enclosed in double quotation marks.

report-delete-from-history-by-instance (rpdfhbi) (report instance ID)

Deletes the report instance from history when the report instance ID is specified.

report-deploy (rpd) (file path) (destination folder) [report type]

Deploys reports to the ZENworks Reporting Server.

The reports are created on the ZENworks Reporting Server with the same name as the report file.

(destination folder) - The folder to which reports are to be deployed.

[report type] - Filters on the type of the reports to be deployed when a folder containing report files is specified. For example, if you specify WID as the report file type, only the Web Intelligence documents in the folder are deployed.

report-folders-sync (rpfs)

Synchronizes the report folders in ZENworks Reporting Server with ZENworks.

report-generate-now (rpgn) (report name) (report folder) (report type) (format type) (report destination) [options]

Generates a report in the specified file format.

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

(format type) - The format in which the report should be exported to the file. Valid values for the `WID` report type are `EXCEL`, `WEBI`, or `PDF`.

(report destination) - The destination of the exported report. Valid values are `BO`, `SMTP`, and `UNMANAGED_DISK`.

Accepts the following options:

-s, --senderAddress=[sender email address] - The e-mail address from which the report should be mailed. This option must be specified if the report destination is SMTP.

-f, --file=[file path] - The path of the file to which the report should be exported. This option must be specified if the report destination is UNMANAGED_DISK.

-u, --username=[username] - The username to access the destination file if it is in a shared location. This option must be specified if the report destination is UNMANAGED_DISK.

-p, --password=[password] - The password to access the destination file if it is in a shared location. This option must be specified if the report destination is UNMANAGED_DISK.

report-instance-export (rpie) (report name) (report folder) (report type) (destination folder) (export format) [options]

Exports the instances of a report in the specified format.

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

(destination folder) - Path to the folder to which the exported report instance files should be saved. The name of the exported file is in the format *report name_report instance id.file type*.

(export format) - The format in which the report should be exported. The valid value is `HTML`.

Accepts the following options:

-s, --start-time=[start time] - Filters the report instances generated after a given time. Specify the time in the format of YYYY-MM-DD HH:MM enclosed in double quotation marks.

-e, --end-time=[end time] - Filters the report instances generated before a given time. Specify the time in the format of YYYY-MM-DD HH:MM enclosed in double quotation marks.

-i, --instance-id=[report instance ID][...] - Displays the ID of the report instances to be exported. If this option is not specified, all the instances of the report are exported. If the `--start-time` or `--end-time` options are specified, report instances generated during the specified time interval are exported.

report-list (rpl) (report type) [options]

Lists reports.

(report type) - Type of the report.

Accepts the following options:

-r, --recursive - Lists results recursively, including subfolders. If this option is not specified, the contents of the subfolders are not listed.

-n, --namefilter=[report name] - Name of the report.

-f, --folder=[folder] - Lists the content of the specified folder. If this option is not specified, the content of the root folder is listed.

report-list-history (rplh) [options] (report name) (report folder) (report type)

Lists the report instances from the history.

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

Accepts the following options:

-s, --start-time=[start time] - Filters the report instances generated after a given time. Specify the time in the format of YYYY-MM-DD HH:MM enclosed in double quotation marks.

-e, --end-time=[end time] - Filters the report instances generated before a given time. Specify the time in the format of YYYY-MM-DD HH:MM enclosed in double quotation marks.

report-load (rpld) (source folder)

Loads all the reports, reporting rights, and ownership details from the specified folder to the Custom Reports folder on the local ZENworks Reporting Server.

(source folder) - Location of the folder from which all the reports and reporting rights must be uploaded. By default, it is the current folder.

report-move (rpmv) (report name) (report folder) (report type) (destination folder)

Moves a report to a different folder.

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

(destination folder) - The folder to which the report should be moved.

report-save (rpsv) (destination folder)

Saves the reports, reporting rights, and the ownership details from the local ZENworks Reporting Server to the specified folder. The reporting rights are saved in the rights.xml file within the specified folder. The ownership details of objects such as Folders or Reports created in the ZENworks Reporting Server InfoView are saved in the ownership.xml file within the specified folder.

The rights.xml file is created only if there are any administrators in the Management Zone. The ownership.xml file is created only if there are any objects created by administrators in the ZENworks Reporting Server InfoView.

(destination folder) - Location of the folder where the reports and reporting rights must be saved. By default, the report and reporting rights are saved in the current folder.

IMPORTANT: The Report-save command (rpsv) saves only the reports that are located in the Custom Reports folder. To save the reports that are located in the Favorites folder, copy the reports to the Custom Reports folder, then run the rpsv command.

report-schedule (rps) (report name) (report folder) (report type) (report destination)

Schedules a report to run now.

(report folder) - Path to the folder containing the report.

(report type) - Type of the report.

(report destination) - The destination of the scheduled report. Valid values are INBOX and BO.

report-users-sync (rpus)

Synchronizes the report users in ZENWorks Reporting Server with ZENworks.

`report-view (rpv) (report ID) (-s|--schedule) (-q|--sql)`

Displays the schedule and SQL query of the report

(-s|--schedule) - Lists the schedule of the report.

(-q|--sql) - Lists the SQL query of the report.

`report-view-settings (rpvs) (-e|--email)`

Displays the report settings, including the e-mail notification information for the reports.

(-e|--email) - E-mail notification information for the report.

Role Commands

These commands are used to create and manage roles for rights-based management of administrator accounts. Role commands begin with `role-` in the long form or the letters `rl` in the short form.

`role-add-admins (rlaa) [role name] [ZENworks administrator name] [...] [rights category] [object path] [...] [options]`

Assigns a role to one or more ZENworks administrators and associates the rights categories in a role to different object contexts.

[rights category] - Name of the category within which the rights are grouped. To see the list of categories, run `zman role-rights-add-admins --Help | more`.

[object path] [...] - Full path of the ZENworks object starting with a slash (/) on which the rights grouped in a rights category should be enforced. To view the root folders of the object types that can be associated with a rights category, run `zman role-add-admins --Help | more`.

Accepts the following options:

-f, --folder=[folder name] - Full path of a folder. If this option is specified, the path of the objects is determined relative to this folder. This provides convenience if you want to specify multiple objects under the same folder.

-H, --Help - Displays detailed help about the categories of rights available, and the root folders of the object types on which the rights can be enforced.

`role-create (rlc) (role name)`

Creates a ZENworks administrator role.

role-delete (rld) (role name) [...]

Renames a ZENworks administrator role.

(role name) [...] - Name of the roles. The wildcard * can be used in the object names if it is enclosed in quotation marks. Exercise caution when using wildcards to delete objects.

role-list (rll) [options]

Lists all ZENworks administrator roles.

Accepts the following options:

-n, --namefilter=[*filter string*] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.

-N, --name-only - Displays the role name only.

-T, --name-and-type - Displays the role name and privilege types only.

-l, --longnames - Displays names of the rights. By default, the abbreviated form of the rights name is displayed.

role-list-admins (rlla) [role name]

Lists the ZENworks administrators with the specified role, and the contexts associated with the rights categories of that role.

role-remove-admins (rlra) (role name) (ZENworks administrator name) [...] [rights category] [object path] [...] [options]

Removes assignment of a role from one or more ZENworks administrators or removes the association of rights categories to object contexts.

[rights category] - Name of the category within which the rights are grouped. To see the list of categories, run `zman role-rights-add-admins --Help | more`.

[object path] [...] - Full path of the ZENworks object starting with / on which the rights grouped in a rights category should be enforced. To view the root folders of the object types that can be associated with a rights category, run `zman role-add-admins --Help | more`.

Accepts the following option:

-f, --folder=[folder name] - Full path of a folder. If this option is specified, the path of the objects is determined relative to this folder. This provides convenience if you want to specify multiple objects under the same folder.

role-rename (rlr) (role name) (new name)

Renames a ZENworks administrator role.

role-rights-delete (rlrd) (role name) [rights category]

Deletes rights assigned to a ZENworks administrator role.

[rights category] - Name of the category within which the rights are grouped. To see the list of categories, run `zman role-rights-set --Help | more`.

role-rights-info (rlri) (role name) [options]

Displays rights assigned to a ZENworks administrator role.

Accepts the following options:

-c, --category=[rights category] - Name of the category within which the rights are grouped. To see the list of categories, run `zman role-rights-set --Help | more`.

-l, --longnames - Displays names of the rights. By default, the abbreviated form of the rights name is displayed.

`role-rights-set (rlrs) [role name] [rights category] [options]`

Allows or denies rights for a ZENworks administrator role.

[rights category] - Name of the category within which the rights are grouped. To see the list of categories, run `zman role-rights-set --Help | more`.

Accepts the following options:

-a, --allow=[right][...] - A comma-separated list of long or short names of the rights to be allowed. To view the rights of each category, run `zman role-rights-set --Help | more`.

-A, --allow-all - Allows all rights of the rights category.

-d, --deny=[right][...] - A comma-separated list of long or short names of the rights to be denied. To view the rights of each category, run `zman role-rights-set --Help | more`.

-D, --deny-all - Denies all rights of the rights category.

-u, --unset=[right][...] - A comma-separated list of long or short names of the rights to be unset. To view the rights of each category, run `zman role-rights-set --Help | more`.

-H, --Help - Displays detailed help about the categories of rights available, and the long and short names of the rights within each category. If a category name is specified, help is provided on the rights for that category.

Ruleset Commands

These commands are used to create and modify registration rule sets. Rule sets are applied to new devices registering to the ZENworks server that do not present a registration key. The commands require a device type to be specified, then the command determines the type of device folder and device group the rule set is associated with, and therefore determines the type of the registering devices. Rule set commands begin with `ruleset-` in the long form, or with the letter `rs` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

ruleset-add-device-groups (rsadg) (device type) (rule set name or position) (device group path) [...] [options]

Adds membership in the specified device groups for devices imported by using the specified rule set.

(device type) - Type of the registering device. Valid values are `server` and `workstation`.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(device group path) - Specify the device group path relative to the root folder of the device type.

Accepts the following option:

-f, --folder=[device folder] - The path of the device folder relative to root folder of device type specified. If this option is specified, the path of the device objects is determined relative to this folder. This can be used to specify multiple device objects under the same folder.

ruleset-add-rule (rsar) (rule set name or position) (add mode) (rule group position) (rules file)

Adds rules or rule groups to a rule set.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(add mode) - Specify the mode of addition, whether rules or rule groups are being added. Valid values are `rule` and `rulegroup`. If mode is `rule`, all rules across rule groups in the input file will be picked and added.

(rules group position) - Specify the position of the rule group. If you are adding rule groups, this refers to the position the new groups must be inserted at. If you are adding rules, it refers to the position in the rule group where the new rule needs to be added. To point to the last rule group, specify *last*. To add rules in a new rule group or to add rule groups at the end of the list, specify *makenew*. The first rule group is at position 1.

(rules file) - The path of the XML file containing the information of rules or rule groups to be added to the rule set. The XML file can be created by exporting rules of an existing rule set using `ruleset-export-rules-to-file` command. The conjunction specified in the XML file is immaterial. Alternately, to add only rules, the older format can still be used.

For the file format, refer to `AddRulesToRuleSet.txt` located in `/opt/novell/zenworks/ share/zman/samples/rulesets` on a Linux Server or `<Installation directory>:\Novell\Zenworks\share\zman\samples\rulesets` on a Windows Server.

For example:

To add rule groups at position 4 in a rule set:

`zman rsar ruleset2 rulegroup 4` *C:\RuleGroups.xml*

To add rules to the third rule group of a rule set:

`zman rsar ruleset2 rule 3` *C:\Rules.xml*

To add rules to the last rule group of a rule set:

`zman rsar ruleset2 rule last` *C:\Rules.xml*

To add rules as a new rule group in a rule set:

`zman rsar ruleset2 rule makenew` *C:\Rules.xml*

ruleset-create (rsc) (device type) (rule set name) (rules file) [position] [options]

Creates a new rule set to apply when registering a device without a registration key.

(device type) - Type of the registering device. Valid values are `server` and `workstation`.

(rule set name) - Specify the name of the rule set.

(rules file) - The path of the XML file containing the information of rules or rule groups for the new rule set. The XML file can be created by exporting rules of an existing rule set by using the `ruleset-export-rules-to-file` command. Alternately, to add only rules (conjunction being AND), the older format can still be used.

For a sample of the file format, see the `AddRulesToRuleSet.txt` file located in `/opt/novell/zenworks/share/zman/samples/rulesets` on a Linux server or *Installation_directory*:`\Novell\Zenworks\share\zman\samples\rulesets` on a Windows server.

(position) - Position of the rule set among the rule sets. The first rule set is at position 1.

Accepts the following options:

-f, --devicefolder=[device folder] - The path of the device folder relative to the root folder of the device type specified. Registering devices are placed in this folder.

-g, --devicegroup=[device group path][...] - A comma-separated list of paths of the device groups. The paths specified should be relative to the root folder of the device type specified. A registered device becomes a member of these device groups.

--desc=[description] - Description for the rule set.

--site=[site] - The site where the devices are located. Registering devices are populated with this site information.

--dept=[department] - The department in which the devices are used. Registering devices are populated with this department information.

--loc=[location] - The physical location of the devices. Registering devices are populated with this location information.

For example:

To create a rule set for registering servers:

```
zman rsc server ruleset1 C:\RuleSet1.xml --devicefolder serverfolder --desc "rule
to import servers" --site "Building A" --dept Finance --loc Brazil
```

Any registering device that evaluates to true for the rules defined in this rule set is placed in the `/Devices/Servers/serverfolder` folder with the given site, department, and location values.

To create a rule set for registering workstations that might become members of some workstation groups:

```
zman rsc workstation ruleset3 --devicegroup wsgroup,wsgroup1,wsfolder/wsgroup2
```

The association of device groups to a rule set can also be done after the rule set creation by using the `ruleset-add-device-group` command.

ruleset-copy (rscp) (rule set name or position) (new name) [position]

Copies a ruleset.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(new name) - New name to be given to the copied ruleset.

(position) - Position of the rule set among the rule sets. The first rule set is at position 1.

For example:

To copy a rule set to the first position:

```
zman rscp ruleset3 newruleset 1
```

Note: If the new position specified is 0 or greater than the number of rule sets, the rule set is copied to the beginning or the end of the list, respectively.

ruleset-export-rules-to-file (rsertf) (rule set name or position) (XML file path)

Exports the rules of a rule set to a file.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(XML file path) - Specify the file in which the rules will be stored in XML format.

For example:

To export rules from a rule set:

```
zman rsertf ruleset1 C:\ExportedRuleSet.xml
```

ruleset-delete (rsd) (rule set name or position)

Deletes a rule set.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

ruleset-info (rsi) (rule set name or position)

Displays detailed information about a rule set.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

For example:

To display details about the ruleset MyRegRule:

```
zman rsi MyRegRule
```

ruleset-list (rsl)

Lists all rule sets.

ruleset-list-device-groups (rsldg) (rule set name or position)

Lists the device groups associated with a rule set.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

ruleset-move (rsmv) (rule set name or position) (new position)

Changes the position of a rule set.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(new position) - Specify the new position to which the rule set needs to be moved.

For example:

```
zman rsmv ruleset3 1
```

Note: If the new position specified is 0 or greater than the number of rule sets, the rule set is moved to the beginning or the end of the list, respectively.

ruleset-remove-device-groups (rsrdg) (device type) (rule set name or position) (device group path) [...] [options]

Removes the association of device groups to a rule set.

(device type) - Type of the registering device. Valid values are server and workstation.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(device group path) - Path of the device groups relative to the root folder of the device type specified.

Accepts the following option:

-f, --folder=[device folder] - The path of the device folder relative to root folder of the device type specified. If this option is specified, the path of the device objects is determined relative to this folder. This can be used to specify multiple device objects under the same folder.

ruleset-remove-rule (rsrr) (rule set name or position) (rule positions) [...]

Removes rules or rule groups from a rule set.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(rule positions) - Specify the positions of the rules to be removed. The first rule group is denoted as G1 and the first two rules in G1 are denoted as G1:1,2. The position can be either a group or specific rules within a group.

For example:

To remove the second rule group of a rule set:

```
zman rsrr ruleset2 G2
```

To remove specific rule groups and specific rules from different rule groups:

```
 zman rsrr ruleset2 G1:3 G2:2,5,6 G3 G6:4,9 G8
```

ruleset-rename (rsr) (rule set name or position) (new name)

Renames a rule set.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

(new name) - Specify the new name to be given to the rule set.

ruleset-update (rsu) (device type) (rule set name or position) [options]

Updates a rule set.

(device type) - Type of the registering device. Valid values are server and workstation.

(rule set name or position) - Specify the name of the rule set or its position within the rule sets. The first rule set is at position 1.

Accepts the following options:

-f, --devicefolder=[device folder] - The path of the device folder relative to the root folder of the device type specified. Registering devices are placed in this folder.

--desc=[description] - Description for the rule set.

--site=[site] - The site where the devices are located. Registering devices are populated with this site information.

--dept=[department] - The department in which the devices are used. Registering devices are populated with this department information.

--loc=[location] - The physical location of the devices. Registering devices are populated with this location information.

For example:

To update a rule set that registers servers:

```
zman rsu server ruleset1 --devicefolder serverfolder1 --desc "rule to import
servers" --site "Building B" --dept Transport --loc Brazil --limit 20
```

To modify server groups in which the imported server should become a member use, `ruleset-add-device-group` and `ruleset-remove-device-group` commands.

To update the device folder from a workstation folder to a server root folder:

```
zman rsu server ruleset2
```

This command changes the folder to which the registering device needs to be placed; from `/Devices/Workstations` to `/Devices/Servers`.

Satellite Commands

These commands are used to view, create, and remove Satellites. Satellite commands begin with `satellite-server-` ![Prema: removing the "ss" shortform as "ss" is being used for another function] .

- "satellite-server-add-content-replication-setting (ssacrs) (path to device) (content type) [-s|--scheduleFileName=file path] [-t|--throttleRate=Throttle Rate] [-d|--duration=Duration] [--noSchedule]" on page 71
- "satellite-server-add-imagingrole (ssaimg) (managed device) [-p|--proxydhcp=ProxyDHCP service]" on page 72
- "satellite-server-create (ssc) (path to device) ((role),(role),...(role)) [parent primary server] [satellite server port] [satellite server secure HTTPS port] [--force-port-change] [--force-parent-change]" on page 72
- "satellite-server-delete (ssd) (Managed device) [(role),(role),...(role)]" on page 73
- "satellite-server-configure-authentication-user-source (sscaus) (path to device) (user source) [user connection, user connection,...,user connection]" on page 73
- "satellite-server-edit-imagingrole (sseimg) (managed device) (action on proxydhcp service)" on page 73
- "satellite-server-export-content (ssec) (path to device) (export directory) [options]" on page 74
- "satellite-server-export-content-replication-setting-schedule (ssecrss) (path to device) (content type) (file path)" on page 74
- "satellite-server-list-authentication-settings (sslas) (path to device)" on page 74
- "satellite-server-list-imagingservers (sslimg)" on page 74
- "satellite-server-list-content-replication-settings (sslcrs) (path to device)" on page 74
- "satellite-server-list-roles (sslr) [managed device]" on page 74
- "Satellite-server-list-servers (ssls)" on page 74
- "satellite-server-remove-content-replication-setting (ssrcrs) (path to device) (content type)" on page 74
- "satellite-server-remove-imagingrole (ssrimg) (managed device) [-r|--removeImageFiles]" on page 75
- "satellite-server-update-content-replication-setting (ssucrs) (path to device) (content type) (file path) [options]" on page 75

satellite-server-add-content-replication-setting (ssacrs) (path to device) (content type) [-s|--scheduleFileName=file path] [-t|--throttleRate=Throttle Rate] [-d|--duration=Duration] [--noSchedule]

Add a content type replication schedule and throttle setting to the Satellite device.

(Path To Device) - The full name of the satellite device to which to add the replication setting.

(Content Type) - The name of the content type for the replication setting to add to the satellite device.

-s, --scheduleFileName=<file path> - The full path to a file with the schedule data for the replication setting saved in XML format.

-t, --throttleRate=<throttle rate> - The throttle rate in kbps for the replication setting to add to the satellite device.

-d, --duration=<duration> - The content replication duration period (in minutes).

--noSchedule - Use this option if you do not want to specify a schedule for content replication.

satellite-server-add-imagingrole (ssaimg) (managed device) [-p|--proxydhcp=ProxyDHCP service]

Promotes a managed device to a Satellite with the Imaging server role. If the managed device is already functioning as a Satellite, this command adds the Imaging server role.

Accepts the following options:

(Managed device) - The path of the managed device relative to /Devices/Workstations.

-p, --proxydhcp=<ProxyDHCP Service> - Starts or stops the Proxy DHCP service. The valid values are start and stop.

NOTE: Before promoting a managed device as Satellite, ensure that the ZENworks version installed on the managed device is same as that of the Primary Server. Additionally, you cannot promote the following devices as a Satellite:

◆ A managed device that has a previous version of ZENworks Adaptive Agent (version 10.2.x or 10.3.x) installed.

◆ A ZENworks 11 SP2 test device.

---

satellite-server-create (ssc) (path to device) ((role),(role),...(role)) [parent primary server] [satellite server port] [satellite server secure HTTPS port] [--force-port-change] [--force-parent-change]

Allows you to configure a Satellite role.

Accepts the following options:

(path to device) - The full name of the managed device to create as a Satellite.

(role) - The Satellite roles to add to the managed device. Roles include Authentication, Collection, and Content If you want to assign the Imaging Satellite role or to promote a managed device to a Satellite with the Imaging server role, you must use the zman ssaimg command instead of this command. If you want to reconfigure JoinProxy role settings, you must use the zman ssujs command.

[Parent Primary Server] - The full name of the Primary Server to which the Satellite rolls up its collection data or from which it obtains its content. This field is optional if the managed device is already a Satellite.

[Satellite Server Port] - The port that the Satellite should use to listen for collection roll-up requests or for obtaining content from the parent Primary Server. If it is not specified, port 80 is used.

[Satellite Server Secure HTTPS Port] - The port that the Satellite should use to listen for authentication secure HTTPS requests. If not specified, port 443 is used.

[--force-port-change] - If the managed device is already functioning as a Satellite, this option allows you to change the port that all Satellite components on this device listen to for requests or for obtaining content.

[--force-parent-change] - If the managed device is already functioning as a Satellite, this option allows you to change the Primary Server parent for all Satellite components on the device.

NOTE: Before promoting a managed device as Satellite, ensure that the ZENworks version installed on the managed device is same as that of the Primary Server. Additionally, you cannot promote the following devices as a Satellite:

- ◆ A managed device that has a previous version of ZENworks Adaptive Agent (version 10.2.*x* or 10.3.*x*) installed.
- ◆ A ZENworks 11 SP2 test device.

A non-root user can successfully execute this command (`zman ssc`) on a Linux Primary Server only if the user has been added to the ZENworks user group. To add the non-root user to the ZENworks user group, use the `usermod -A non-root_username` command.

satellite-server-delete (ssd) (Managed device) [(role),(role),...(role)]

Allows you to delete Satellite roles from the device. If all roles are removed, the device will be automatically delisted from the Server Hierarchy listing in ZENworks Control Center.

If your Management Zone consists of ZENworks 11 SP2 Primary Server and ZENworks Configuration Management 10.2.*x*/10.3.*x* Satellites, you cannot remove individual roles from the Satellites. You can only demote the Satellite to a managed device. You can delete a Satellite that has any version of the ZENworks Adaptive Agent installed.

Accepts the following options:

(Managed device) - The full name of the managed device from which to delete the Satellite roles.

[role] - The Satellite roles to delete from the device. If no roles are specified, all Satellite roles are deleted from the device. You can delete the following roles: Authentication, Collection, and Content

. To remove the Imaging role, use the `satellite-server-remove-imagingrole (ssrimg)` command instead of this command.

satellite-server-configure-authentication-user-source (sscaus) (path to device) (user source) [user connection, user connection,...,user connection]

Configures the user source connections used by a Satellite with the Authentication role. Execute this command multiple times to configure connections for multiple user sources on the same Satellite, specifying a different user source each time. Execute this command with no connections to remove the specified user source from the authentication role of the Satellite.

Accepts the following options:

(Path To Device) - The full name of the Satellite for which to configure the authentication role user source.

(User Source) - The name of the user source to configure.

[User Connection] - User connections listed in order that this Satellite uses to authenticate users against the given user source.

satellite-server-edit-imagingrole (sseimg) (managed device) (action on proxydhcp service)

Starts or stops the Proxy DHCP service of the Imaging Satellite.

(Managed device) - The path of the managed device relative to `/Devices/Workstations`. The device specified must be a Satellite with the Imaging role.

(Action on ProxyDHCP Service) - Starts or stops the Proxy DHCP service. The valid values are `start` and `stop`.

satellite-server-export-content (ssec) (path to device) (export directory) [options]

Exports content files from the content repository for manual import into the Satellite device's content repository. To import the content into the content repository on a managed device, use the `zac cdp-import-content (cic)` command.

(Path To Device) - The full name of the Satellite device for which to export the content files.

(Export Directory) - The full path to a directory to which to export the content files.

Accepts the following options:

-a, --exportAll - Exports all content assigned to the Satellite device (by default, only missing content is exported).

satellite-server-export-content-replication-setting-schedule (ssecrss) (path to device) (content type) (file path)

Export the content type replication schedule from the Satellite device to a file.

(Path To Device) - The full name of the satellite device from which to export the replication setting schedule.

(Content Type) - The name of the content type for the replication setting to export from the satellite device.

(file path) - The full path to a file to which to export the schedule data.

satellite-server-list-authentication-settings (sslas) (path to device)

Lists the authentication role settings on a Satellite device.

(path to device) - The full name of the satellite device from which to list the authentication role settings.

satellite-server-list-imagingservers (sslimg)

Lists the Satellites with the Imaging role.

satellite-server-list-content-replication-settings (sslcrs) (path to device)

Lists the content replication schedule and throttle settings on the Satellite by content type.

(path to device) - The full name of the satellite device from which to list the replication settings.

satellite-server-list-roles (sslr) [managed device]

Lists previously defined Satellite roles.

Accepts the following option:

[Managed device] - The full name of the managed device for which to list Satellite roles. If not specified, all available Satellite roles are displayed.

Satellite-server-list-servers (ssls)

Lists the Satellites and their roles.

satellite-server-remove-content-replication-setting (ssrcrs) (path to device) (content type)

Remove a content type replication schedule and throttle setting from the Satellite device.

(Path To Device) - The full name of the satellite device from which to remove the replication setting.

(Content Type) - The name of the content type for the replication setting to remove from the satellite device.

```
satellite-server-remove-imagingrole (ssrimg) (managed device) [-r|--
removeImageFiles]
```

Removes the Satellite Imaging server role from the managed device.

If your Management Zone consists of ZENworks 11 SP2 Primary Server and ZENworks Configuration Management 10.2.*x*/10.3.*x* Satellites, you cannot remove individual roles from the Satellites. You can only demote the Satellite to a managed device. You can delete a Satellite that has any version of the ZENworks Adaptive Agent installed.

(Managed device) - The path of the managed device relative to `/Devices/Workstations`. The device specified must be a Satellite with the Imaging role.

-r, --removeImageFiles - Deletes the image files from the specified device.

```
satellite-server-update-content-replication-setting (ssucrs) (path to device)
(content type) (file path) [options]
```

Updates a content type replication schedule and throttle setting to the Satellite device.

(Path To Device) - The full name of the satellite device to which to update the replication setting.

(Content Type) - The name of the content type for the replication setting to update to the satellite device.

(file path) - The full path to a file with the schedule data for the replication setting saved in XML format.

Accepts the following option:

-d, --duration=<duration> - The content replication duration period (in minutes).

-t, --throttleRate=<Throttle Rate> - The throttle rate in kbps for the replication setting to update to the satellite device.

Server Commands

These commands are used to manage servers. Server commands begin with the word `server-` in the long form, or with the letter `s` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

- "server-add-bundle (sab) (server object path) (bundle or bundle group) [...] [options]" on page 76
- "server-add-certificate (sacert) (server object path) (certificate file path)" on page 77
- "server-add-policy (sap) (server object path) (policy or policy group) [...] [options]" on page 77
- "server-delete (sd) (server object path) [...] [options]" on page 78
- "server-folder-create (sfc) [folder name] [parent folder] [options]" on page 78
- "server-group-add (sga) (server group path) [server path] [...] [options]" on page 78
- "server-group-create (sgc) [group name] [parent folder] [options]" on page 78
- "server-group-members (sgm) (server group path) [options]" on page 78
- "server-group-remove (sgr) (server group path) [server path] [...] [options]" on page 78
- "server-info (si) (server path)" on page 79
- "server-list (sl) [options] [folder]" on page 79
- "server-list-bundles (slb) (server object path) [options]" on page 79
- "server-list-groups (slg) (server path) [options]" on page 79

server-add-bundle (sab) (server object path) (bundle or bundle group) [...] [options]

Assigns bundles to a server object.

Accepts the following options:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

-I, --icon-location=[application location XML file] - XML file that contains the locations to place the icon for the bundle application. For the XML file format, refer to IconLocation.xml located in /opt/novell/zenworks/share/zman/samples/bundles on a Linux server or *Installation_directory*:\Novell\Zenworks\share\zman\samples\ bundles on a Windows server.

-d, --distribution-schedule=[distribution schedule XML file] - The XML file that contains the distribution schedule.

-l, --launch-schedule=[launch schedule XML file] - The XML file that contains the launch schedule.

-a, --availability-schedule=[availability schedule XML file] - The XML file that contains the availability schedule. For the schedule XML file templates, refer to the XML files located in / opt/novell/zenworks/share/zman/samples/schedules on a Linux server or *Installation_directory*:\Novell\Zenworks\share\zman\samples\ schedules on a Windows server.

-i, --install-immediately - Installs the bundle immediately after distribution.

To use this option, you must also specify the Distribution schedule. The Distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-L, --launch-immediately - Launches the bundle immediately after installation.

To use this option, you must also specify the Distribution schedule. The Distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-n, --distribute-now - Sets the distribution schedule to distribute the bundle immediately. If this option is specified, the --distribution-schedule and --distribute-on-device-refresh options are ignored. The --distribute-now, --distribute-on-device-refresh, and --distribution-schedule options are mutually exclusive and are used to set the distribution schedule. The --distribute-now option is considered first, followed by --distribute-on-device-refresh and --distribution-schedule.

-r, --distribute-on-device-refresh - Sets the distribution schedule to distribute the bundle on device refresh. If this option is specified, the --distribution-schedule option is ignored.

-s, --launch-on-device-refresh - Sets the launch schedule to launch the bundle on device refresh. If this option is specified, the --launch-schedule option is ignored.

-w, --wakeup-device-on-distribution - Wakes up the device by using Wake-On-LAN if it is shut down while distributing the bundle.

To use this option, you must also specify the distribution schedule. The distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-B, --broadcast=[Broadcast address][...] - A comma-separated list of addresses used to broadcast the Wake-On-LAN magic packets. This option is used only if you choose to wake up the device by using Wake-On-LAN. A valid IP address is a valid value.

-S, --server=[The path of the Primary or Proxy Server objects relative to /Devices][...] - A comma-separated list of Primary or Proxy Server objects used to wake up the device. This option is used only if you choose to wake up the device by using Wake-On-LAN.

-C, --retries=[Number of retries] - Number of times the Wake-On-LAN magic packets are sent to the device(s). This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 0 and 5. The default value is 1.

-T, --timeout=[Time interval between retries] - The time interval between two retries. This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 2 and 10 (in minutes). The default value is 2.

server-add-certificate (sacert) (server object path) (certificate file path)

Adds a second valid certificate for a server in preparation for replacing a certificate that is about to expire

(server object path) - The path of the server objects (server, server folder or server group) relative to /Devices/Servers.

(certificate file path)- Path of the DER format certificate file.

server-add-policy (sap) (server object path) (policy or policy group) [...] [options]

Assigns policies to a server object.

Accepts the following options:

-c, --conflicts=[policy conflict resolution order] - Determines how policy conflicts are resolved. Valid values are userlast or 1, devicelast or 2, deviceonly or 3, useronly or 4. For userlast, device-associated policies are applied first, followed by user-associated policies. For devicelast, user-associated policies are applied first, followed by device-associated policies. For deviceonly, user-associated policies are ignored. For useronly, device-associated policies are ignored. If this option is not specified, userlast is taken as the default value.

-e, --enforce-now - Enforces the policy immediately on all assigned devices.

-f, --folder=[policy folder] - The path of a policy folder relative to `/Policies`. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

server-delete (sd) (server object path) [...] [options]

Deletes one or more server objects.

(server object path) [...] - The path of the server objects (server, server folder or server group) relative to `/Devices/Servers`. The wildcard * can be used in the object names if it is enclosed in quotation marks. Exercise caution while using wildcards for deleting objects.

Accepts the following options:

-r, --recursive - Deletes objects inside a folder recursively.

-f, --folder=[server folder] - The path of a server folder relative to `/Devices/Servers`. If this option is specified, the path of the server objects is determined relative to this folder. This can be used to specify multiple server objects under the same folder.

server-folder-create (sfc) [folder name] [parent folder] [options]

Creates a new folder for containing servers.

Accepts the following option:

--desc=[description] - Description for the folder.

server-group-add (sga) (server group path) [server path] [...] [options]

Adds servers to a server group.

Accepts the following option:

-f, --folder=[server folder] - The path of a server folder relative to `/Devices/Servers`. If this option is specified, the path of the server objects is determined relative to this folder. This can be used to specify multiple server objects under the same folder.

server-group-create (sgc) [group name] [parent folder] [options]

Creates a server group and adds members to it.

Accepts the following options:

--desc=[description] - Description for the group.

-m, --members=[server path][...] - The path of the servers relative to `/Devices/Servers`.

-f, --folder=[server folder] - The path of a server folder relative to `/Devices/Servers`. If this option is specified, the path of the server objects is determined relative to this folder. This can be used to specify multiple server objects under the same folder.

server-group-members (sgm) (server group path) [options]

Lists members of a server group or a dynamic server group.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

server-group-remove (sgr) (server group path) [server path] [...] [options]

Removes servers from a server group.

Accepts the following option:

-f, --folder=[server folder] - The path of a server folder relative to /Devices/Servers. If this option is specified, the path of the server objects is determined relative to this folder. This can be used to specify multiple server objects under the same folder.

server-info (si) (server path)

Lists detailed information about a server.

server-list (sl) [options] [folder]

Lists server objects.

Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, the contents of the subfolders are not listed.

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.

-t, --typefilter=[type filter][...] - Displays results matching the comma-separated list of object types specified. Valid values are device, group, and folder.

-c, --count - Displays the count of the results.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

-f, --flagfilter=[flag filter][...] - Displays results matching the comma-separated list of flags specified. Valid values are retired, notretired, test, and non-test.

server-list-bundles (slb) (server object path) [options]

Lists bundles assigned to a server object.

Accepts the following options:

-a, --all - Lists both effective and non-effective bundles.

-e, --effective - Lists only effective bundles.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

server-list-groups (slg) (server path) [options]

Lists groups of which the given server is a member.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

server-list-policies (slp) (server object path) [options]

Lists policies assigned to a server object.

Accepts the following options:

-a, --all - Lists both effective and non-effective policies.

-e, --effective - Lists only effective policies.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

server-move (smv) (server object path) [destination folder path]

Moves a server object to a different folder.

server-refresh (srf) (server object path) [...] [options]

Refreshes the ZENworks Adaptive Agent in the servers.

Accepts the following option:

-f, --folder=[server folder] - The path of a server folder relative to /Devices/Servers. If this option is specified, the path of the server objects is determined relative to this folder. This can be used to specify multiple server objects under the same folder.

server-remove-bundle (srb) (server object path) (bundle or bundle group) [...] [options]

Removes bundles assigned to a server object.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

server-remove-policy (srp) (server object path) (policy or policy group) [...] [options]

Removes policies assigned to a server object.

Accepts the following option:

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This helps you to specify multiple policy objects under the same folder.

server-rename (sr) (server object path) (new name)

Renames a server object.

server-reorder-bundles (srob) (server object path) (current position) (new position)

Changes the order of bundles assigned to a server. Use the server-list-bundles command to get the order of assigned bundles.

server-reorder-policies (srop) (server object path) (current position) (new position)

Changes the order of policies assigned to a server. Use the server-list-policies command to get the order of assigned policies.

server-retire (sret) (server object path)

Retires the selected server from your ZENworks system at its next refresh. Retiring a device is different from deleting a device. When you retire a device, its GUID is retained (as opposed to when you delete a device, which also deletes its GUID). As a result, all inventory information is retained and is accessible, but all policy and bundle assignments are removed. If you reactivate the device in the future, its assignments are restored.

Accepts the following option:

-i, --immediate - Forces a device refresh to immediately retire the device.

server-set-as-test (ssat) (server object path)

Sets a server as a test device.

Examples

To set server1 as a test device:

zman ssat server1

server-set-as-nontest (ssan) (server object path)

> Sets a server as a non-test device.
>
> Examples
>
> To set server1 as a non-test device:
>
> `zman ssan server1`

server-statistics (sst) (server path)

> Displays statistics of ZENworks server usage.

server-unretire (suret) (server object path)

> Reactivates the selected server at its next refresh and reapplies all policy and bundle assignments that the device previously had.
>
> Accepts the following option:
>
> -i, --immediate - Forces a device refresh to immediately unretire the device.

server-view-all-bundles-status (svabs) (server path) [options]

> Displays the advanced deployment status of bundles assigned to a server.
>
> Accepts the following options:
>
> Accepts the following options:
>
> --statusfilter=[status type][...] - Filter on the status of Bundle Distribution and Install Events. Valid values are S, R, C, A, F, and P (Success, Partial Success, Completed, Partial Complete, Failure, and Pending). A comma-separated list of status types can be given.
> -s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.
>
> Examples
>
> To view the deployment status of bundles assigned to server1:
>
> `zman svabs server1`
>
> To view the deployment status of bundles assigned to server1 having bundle deployment status as success or pending:
>
> `zman svabs server1 --statusfilter S,P`

server-view-all-policies-status (svaps) (server path) [options]

> Displays the advanced deployment status of policies assigned to a server.
>
> Accepts the following options:
>
> --statusfilter=[status type][...] - Filters on the status of Policy Apply Event. Valid values are S, F, and P (Success, Failure, and Pending). A comma-separated list of status types can be given.
> -s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

server-wakeup (sw) (server object path) [...] [options]

> Wakes up a server using Wake-On-LAN.
>
> Accepts the following options:
>
> -f, --folder=[server folder] - The path of a server folder relative to /Devices/Servers. If this option is specified, the path of the server objects is determined relative to this folder. This can be used to specify multiple server objects under the same folder.
> -b, --broadcast=[Broadcast address][...] - A comma-separated list of addresses used to broadcast the Wake-On-LAN magic packets. This option is used only if you choose to wake up the device by using Wake-On-LAN. A valid IP address is a valid value.

-s, --server=[Path of the Primary or Proxy Server objects relative to /Devices][...] - A comma-separated list of Primary or Proxy Server objects used to wake up the device. This option is used only if you choose to wake up the device by using Wake-On-LAN.

-r, --retries=[Number of retries] - Number of times the Wake-On-LAN magic packets are sent to the devices. This option is used only if you choose to wake up the device by using   Wake-On-LAN. The value must be between 0 and 5. The default value is 1.

-t, --timeout=[Time interval between retries] - The time interval between two retries. This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 2 and 10 (in minutes). The default value is 2.

Settings Commands

These commands are used to set and modify settings. Settings commands begin with `settings-` in the long form, or with the letter `s` in the short form.

The Settings commands do not apply to ZENworks Endpoint Security Management settings (Zone Policy settings and Date Encryption Key settings).

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

- "settings-copy (scp) (source device or device folder path) (destination device or device folder path) [...] [options]" on page 82
- "settings-export-to-file (setf) [options] (XML file path) [settings name] [...]" on page 82
- "settings-revert (srt) (source device or device folder path) (settings name) [...]" on page 83
- "settings-set (ss) (XML file path) [destination device or device folder path] [...]" on page 83

settings-copy (scp) (source device or device folder path) (destination device or device folder path) [...] [options]

Copies settings.

(source device or device folder path) - The path of the device or device folder relative to / `Devices` from which settings have to be copied.

(destination device or device folder path) [...] - The path of the devices or device folders relative to /Devices to which settings must be copied.

Accepts the following option:

-n, --names=[settings name][...] - Names of the settings to be copied from the source path. If this option is not specified, all settings defined at the source path are copied.

settings-export-to-file (setf) [options] (XML file path) [settings name] [...]

Exports settings data (in XML format) to a file. The XML file is to be used as input for creating or modifying settings.

(XML file path) - The file into which the settings data is stored in XML format. If the file does not exist, a new file is created.

[settings name] [...] - Names of the settings to be exported. If this option is not specified, all settings are exported.

Accepts the following options:

-s, --source=[source device or device folder path] - The path of the device or device folder relative to /Devices from which settings must be exported. If this option is not specified, settings are exported from the Zone.

-e, --effective - If specified, the effective settings are retrieved; otherwise, only the settings defined or overridden at the source path are retrieved.

settings-revert (srt) (source device or device folder path) (settings name) [...]

Reverts the settings to that of the parent folder.

(source device or device folder path) - The path of the device or device folder relative to / Devices, whose settings must be reverted.

settings-set (ss) (XML file path) [destination device or device folder path] [...]

Sets settings at various levels.

(XML file path) - XML file containing exported settings information. Use settings-export-to-file (setf) to export settings information into a XML file.

Accepts the following options:

[settings name] [...] - Names of the settings to be exported. If this option is not specified, all settings are exported.

-s, --source=[source device or device folder path] - The path of the device or device folder relative to /Devices for which settings have to be set. If this option is not specified, settings are set at the Zone level.

Subscription Commands

These commands are used to create, modify, and manage subscriptions. Subscription commands begin with subscription- in the long form, or with the letters sr in the short form.

This command accepts the option flags listed in the Global Options section. In addition, it accepts individual options, as listed with the command.

- "subscription-create (src) (subscription name) (subscription XML file) [parent folder] [options]" on page 83
- "subscription-folder-create (srfc) (folder name) [parent folder] [options]" on page 84
- "subscription-move (srmv) (subscription object path) [destination folder path]" on page 84
- "subscription-replicate (srr) (subscription-name) [destination folder path] [options]" on page 84
- "subscription-export (sre) (subscription-name) (subscription-data.xml)" on page 84
- "subscription-delete (srd) (Subscription object) […] [options]" on page 84
- "subscription-list (srl) [options]" on page 84
- "subscription-modify (srm) (subscription-name) (subscription-data.xml) [parent folder] [options]" on page 85
- "subscription-status (srs) (subscription-name) [options]" on page 85
- "subscription-channel-list (srcl) (subscription-name) (repo entity type) [options]" on page 85
- "subscription-copy (srcp) (subscription path) (subscription-name) [parent folder] [options]" on page 85
- "subscription-rename (srrn) (subscription object path) (new name)" on page 85

subscription-create (src) (subscription name) (subscription XML file) [parent folder] [options]

Creates a new subscription.

(subscription XML file) - XML file containing exported subscription information. Use `subscription-export (sre)` to export an existing subscription's information into an XML file. If you want to reuse files, template XML files can be created from subscriptions that were created through ZENworks Control Center. For a sample XML file, refer to `nu.xml` located in

`/opt/novell/zenworks/share/zman/samples/subscriptions` on a Linux server.

Accepts the following options:

-d, --desc=[description] - Description for the subscription.

-s, --sysIDpath=[rhel-systemid]

subscription-folder-create (srfc) (folder name) [parent folder] [options]

Creates a new subscription folder.

Accepts the following option:

--desc=[description] - Description for the folder.

subscription-move (srmv) (subscription object path) [destination folder path]

Moves a subscription object to a different folder.

subscription-replicate (srr) (subscription-name) [destination folder path] [options]

Begins a subscription replication.

Accepts the following options:

-s, --schedule=[XML filename] - The XML file that contains the schedule to begin subscription. The `Eventsch.xml` file is not applicable for subscriptions.
-r, --fromRepository - The repository from which to replicate content.

subscription-export (sre) (subscription-name) (subscription-data.xml)

Exports subscription information (in XML format) to a file. The XML file is to be used as input for creating subscriptions.

subscription-delete (srd) (Subscription object) […] [options]

Deletes one or more subscription objects. The subscriptions that are in InProgress, Suspended, and Assigned status cannot be deleted.

Accepts the following options:

-r, --recursive - Deletes objects inside a folder recursively.
-b, --delete-bundles - Deletes all the bundles that were created by the specified subscription.
-f, --folder=[subscription folder] - The path of a subscription folder relative to `/Subscriptions`. If this option is specified, the path of the subscription objects is determined relative to this folder. This option can be used to specify multiple subscription objects under the same folder.

subscription-list (srl) [options]

Lists subscription objects.

Accepts the following options:

-r, --recursive - Lists results recursively, including subfolders. If this option is not specified, the contents of the subfolders are not listed.
-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.
-c, --count - Displays the count of the results.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

[folder] - Lists the contents of the specified folder. If you do not specify a folder, the contents of the root folder are listed.

subscription-modify (srm) (subscription-name) (subscription-data.xml) [parent folder] [options]

Exports or modifies an existing subscription.

Accepts the following options:

-d, --desc=[description] - Description for the subscription.

-s, --sysIDpath=[rhel-systemid]

subscription-status (srs) (subscription-name) [options]

Returns the subscription's status.

Accepts the following options:

-s, --suspend - Suspends the replication and changes the subscription status to Suspended.

-c, --cancel - Cancels the replication and changes the subscription status to Canceled.

NOTE: The Suspend and Cancel tasks are not supported while performing a dry run.

subscription-channel-list (srcl) (subscription-name) (repo entity type) [options]

Lists data such as bundles, catalogs, and packages.

Accepts the following options:

-r, --fromRepository - Bypasses the cache and downloads the metadata from the repository from which to replicate the content.

-i, --ignoreFilters - List the content, ignoring any filters that might be configured.

subscription-copy (srcp) (subscription path) (subscription-name) [parent folder] [options]

Copies a subscription.

Accepts the following option:

-d, --desc=description - Description for the bundle.

subscription-rename (srrn) (subscription object path) (new name)

Renames a subscription.

For example, to rename a subscription SLES10Update to SLED10Update, run the command as follows:

```
zman srrn /Subscription/SLES10Update /Subscription/SLED10Update
```

System Update/Product Recognition Update Commands

These commands are used to manage System Updates and Product Recognition Updates (PRUs) to ZENworks. These commands begin with system-update in the long form, or with the letters su in the short form.

- "system-update-download (sud) (system update or PRU name)" on page 86
- "system-update-view-available (suva) [options]" on page 86
- "system-update-status (sus) (system update or PRU name)" on page 86
- "system-update-delete-update (sudu) (system update or PRU name) [options]" on page 86

system-update-download (sud) (system update or PRU name)

Downloads the latest system updates or PRU.

(system update or PRU name) - System update or PRU name or UID.

system-update-view-available (suva) [options]

Displays a list of all available updates or PRUs ed on the update server or in an update file.

Accepts the following options:

-i, --importFile=[path to import file/directory] - Full path to the import file to query.
-u, --url=[url to update server] - URL for the system update server to query in the format http://servername:[port]/path.
-f, --product=[product code] - The product code to check for updates (for example, zcm, pru).
-v, --version=[product base version] - The base version of the product to check for updates (for example, 10.0, 10.1).

system-update-status (sus) (system update or PRU name)

Displays the status of the specified system update or PRU.

(system update or PRU name) - System update or PRU name or UID.

system-update-delete-update (sudu) (system update or PRU name) [options]

Deletes a specified system update or PRU from the zone.

(system update or PRU name) - System update or PRU name or UID.

Accepts the following option:

-f, --force - Forces the update to be deleted. Use with caution, and only if you are sure that the update is not in an active state.

system-update-cancel-download (sucd) (system update or PRU name)

Cancels the download of the system update or PRU.

(system update or PRU name) - System update or PRU name or UID.

system-update-assign (sua) (system update name) [device path] [...] [option]

Assigns the authorized system update to devices.

(system update name) - System update name or UID.

[device path] [...] - The path of the device relative to /Devices.

Accepts the following option:

-a, --all - Assign to all devices.

`system-update-list-assign (sula) (device path)`

Lists all updates assigned to the specified device or group.

(device path) - The path of the device relative to `/Devices`.

`system-update-authorize (suaz) (system update or PRU name)`

Authorizes an update or PRU.

(system update or PRU name) - System update or PRU name or UID.

`system-update-baseline (sub) (system update name)`

Schedules an attempt to set the specified update as the baseline update.

(system update name) - System update name or UID.

`system-update-import (sui) (path to update) [system update or PRU name] [-b,--checkbaseline]`

Imports a system update or PRU from a ZIP file or directory into the Zone. After the update or PRU is in the ZENworks system, it can be managed the same way that online updates or PRUs are managed.

(path to update) - The full path to the update or PRU ZIP file or directory.

(system update or PRU name) - System update or PRU name or UID.

(-b, --checkbaseline) - Check for and if applicable, baseline the update. The default is to not baseline the update.

`system-update-activate-entitlement (suae) (mail ID) (activation code)`

Activates the System Update entitlement for the ZENworks 11SP2 software in the Management Zone to receive the latest version of ZENworks System Updates and Product Recognition Updates (PRUs) from the Novell Customer Center (NCC) server.

(mail ID) - A valid e-mail address to be used for communication from Novell. We recommended that you specify the e-mail address used to purchase the System Update Maintenance Entitlement.

(activation code) - The System Update entitlement activation code.

NOTE: Before activating the entitlement, ensure that the Primary Server you want to use to activate the entitlement can communicate with the NCC server (https://secure-www.novell.com).

`system-update-view-entitlement-status (suves)`

Displays the System Update entitlement status.

`system-update-rebuild-packages (surp) (system update name)`

Rebuilds the deployment packages on this server with the content of the specified update.

(system update name) - System update name or UID.

`system-update-patch-file (supf) (system update name) (path to patch)`

Patches an update with new versions of file(s).

(system update name) - System update name or UID.

(path to patch) - Full path of the zip file or the folder where patch files can be found.

User Commands

These commands are used to manage users, belonging to the user sources configured in ZENworks. The user sources are to be configured from an existing LDAP Authoritative Source like e-Directory or Active Directory. User commands begin with user- in the long form, or with the letter u in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

user-add-bundle (uab) (user object path) (bundle or bundle group) [...] [options]

Assigns bundles to a user object.

Accepts the following options:

-f, --folder=[bundle folder] - The path of a bundle folder relative to `/Bundles`. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

-I, --icon-location=[application location XML file] - XML file that contains the locations to place the icon for the bundle application. For the XML file format, refer to `IconLocation.xml` located in `/opt/novell/zenworks/share/zman/samples/bundles` on a Linux server or *Installation_directory*:`\Novell\Zenworks\share\zman\samples\` `bundles` on a Windows server.

-d, --distribution-schedule=[distribution schedule XML file] - XML file that contains the distribution schedule.

-l, --launch-schedule=[launch schedule XML file] - XML file that contains the launch schedule.

-a, --availability-schedule=[availability schedule XML file] - XML file that contains the availability schedule. For the schedule XML file templates, refer to the XML files located in `/` `opt/novell/zenworks/share/zman/samples/schedules` on a Linux server or *Installation_directory*:`\Novell\Zenworks\share\zman\samples\` `schedules` on a Windows server.

-i, --install-immediately - Installs the bundle immediately after distribution.

To use this option, you must also specify the distribution schedule. The distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-L, --launch-immediately - Launches the bundle immediately after installation.

To use this option, you must also specify the distribution schedule. The distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-n, --distribute-now - Sets the distribution schedule to distribute the bundle immediately. If this option is specified, the --distribution-schedule and --distribute-on-device-refresh options are ignored. The --distribute-now, --distribute-on-device-refresh, and --distribution-schedule options are mutually exclusive and are used to set the distribution schedule. The --distribute-now option is considered first, followed by --distribute-on-device-refresh and --distribution-schedule.

-r, --distribute-on-device-refresh - Sets the distribution schedule to distribute the bundle on device refresh. If this option is specified, the --distribution-schedule option is ignored.

-s, --launch-on-device-refresh - Sets the launch schedule to launch the bundle on device refresh. If this option is specified, the --launch-schedule option is ignored.

user-add-policy (uap) (user object path) (policy or policy group) [...] [options]

Assigns policies to a user object.

Accepts the following options:

-e, --enforce-now - Enforces the policy immediately on all assigned devices.

-f, --folder=[policy folder] - The path of a policy folder relative to `/Policies`. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

user-container-create (ucc) (user source) (user container) (display name)

Adds a user container to a user source configured in ZENworks.

(user container) - The user container to be added in RDN (Relative Distinguished Name) format, relative to the root context of the LDAP Source.

(display name) - Displays the name for the user container.

user-group-add (uga) (ZENworks user group path) (user path) [...] [options]

Adds users to a ZENworks user group.

Accepts the following option:

-f, --folder=[user folder] - The path of a user folder relative to /Users. If this option is specified, the path of the user objects is determined relative to this folder. This can be used to specify multiple user objects under the same folder.

user-group-create (ugc)(group name)[options]

Creates a ZENworks user group and adds members to it.

Accepts the following options:

--desc=[description] - Description for the group.

-m, --members=[user path][...] - The path of the users relative to /Users.

-f, --folder=[user folder] - The path of a user folder relative to /Users. If this option is specified, the path of the user objects is determined relative to this folder. This can be used to specify multiple user objects under the same folder.

user-group-delete (ugd) (ZENworks user group path) [...]

Deletes one or more ZENworks user groups.

(ZENworks user group path) [...] - The path of the ZENworks user group relative to /Users/ ZENworks User Groups.

user-group-members (ugm) (user group path) [options]

Lists members of a ZENworks user group or LDAP Directory user group.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

user-group-remove (ugr)(ZENworks user group path) (user path) [...] [options]

Removes users from a ZENworks user group.

Accepts the following option:

-f, --folder=[user folder] - The path of a user folder relative to /Users. If this option is specified, the path of the user objects is determined relative to this folder. This can be used to specify multiple user objects under the same folder.

user-info (ui) (user path)

Lists detailed information about a user.

user-list (ul) [options] [folder]

Lists user objects.

Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, the contents of the subfolders are not listed.

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.

-t, --typefilter=[type filter][...] - Displays results matching the comma-separated list of object types specified. Valid values are `user`, `group`, and `folder`.

-c, --count - Displays the count of the results.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`user-list-bundles (ulb) (user object path) [options]`

Lists bundles assigned to a user object.

Accepts the following options:

-a, --all - Lists both effective and non-effective bundles.

-e, --effective - Lists only effective bundles.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`user-list-groups (ulg) (user path) [options]`

Lists groups of which the given user is a member.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`user-list-policies (ulp) (user object path) [options]`

Lists policies assigned to a user object.

-a, --all - Lists both effective and non-effective policies.

-e, --effective - Lists only effective policies.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`user-logged-in-devices (ulid) (user path)`

Displays the devices on which a user is logged in.

`user-primary-workstation (upws) (user path)`

Lists the user's primary workstation, number of logins, and number of minutes that the user has been logged in to the primary workstation.

`user-remove-bundle (urb) (user object path) (bundle or bundle group) [...] [options]`

Removes bundles assigned to a user object.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to `/Bundles`. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

`user-remove-policy (urp) (user object path) (policy or policy group) [...] [options]`

Removes policies assigned to a user object.

Accepts the following option:

-f, --folder=[policy folder] - The path of a policy folder relative to `/Policies`. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

`user-reorder-bundles (urob) (user object path) (current position) (new position)`

Changes the order of bundles assigned to a user.

user-reorder-policies (urop) (user object path) (current position) (new position)

Changes the order of policies assigned to a user.

user-source-add-connection (usac) (user source) (connection name) (server address) (server port) [options]

Adds a new server connection for a user source.

Accepts the following options:

-a, --accept-certificate - Accepts the certificate presented by the user source when SSL mode is chosen. This option is provided to avoid interactive mode for scripting. It is advisable to view the certificate and then accept it.

-s, --use-ssl - Specifies that SSL should be used when accessing the new server.

user-set-as-test (usat) (user object path)

Sets an user as a test user.

Examples

To set user1 as a test user:

```
zman usat user1
```

user-set-as-nontest (usan) (user object path)

Sets an user as a non-test user.

Examples

To set user1 as a non-test user:

```
zman usan user1
```

user-source-create (usc) (user source XML file path) [options]

Configures a user source.

Accepts the following option:

-a, --accept-certificate - Accepts the certificate presented by the user source when SSL mode is chosen. This option is provided to avoid interactive mode for scripting. It is advisable to view the certificate and then accept it.

user-source-list (usl)

Lists the configured user sources.

user-source-list-connections (uslc) (user source)

List the server connections for a user source.

user-source-remove-connection (usrc) (user source) (connection name)

Removes a server connection from a user source.

user-source-trustedcert-add (usta) (certificate alias) (certificate file path)

Adds an additional trusted certificate for another LDAP server that can be referred to during the processing of an LDAP search request when using secure user source connections.

(certificate alias) - The alias used to identify the certificate in the key store.
(certificate file path) - The path to the DER format certificate file.

user-source-trustedcert-list (ustl)

Lists the additional trusted certificates used for secure user source connections.

`user-source-trustedcert-remove` **(user-source-trustedcert-remove|ustr) (certificate alias)**

Removes a trusted certificate used for secure user source connections.

(certificate alias) - The alias used to identify the certificate in the key store.

`user-view-all-bundles-status (uvabs) (user path) [options]`

Displays the advanced deployment status of bundles assigned to a user.

Accepts the following options:

--statusfilter=[status type][...] - Filters on the status of Bundle Distribution and Install Events. Valid values are S, F, and P (Success, Failure, and Pending). A comma-separated list of status types can be given.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

`user-view-all-policies-status (uvaps) (user path) [options]`

Displays the advanced deployment status of policies assigned to a user.

Accepts the following options:

--statusfilter=[status type][...] - Filters on the status of Policy Apply Event. Valid values are S, F, and P (Success, Failure, and Pending). A comma-separated list of status types can be given.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

Workstation Commands

These commands are used to manage workstations. Workstation commands begin with `workstation-` in the long form, or with the letter `w` in the short form.

All of the commands below accept the option flags listed in the Global Options section. In addition, they accept individual options as listed with each command.

workstation-add-bundle (wab) (workstation object path) (bundle or bundle group) [...] [options]

Assigns bundles to a workstation object.

Accepts the following options:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

-I, --icon-location=[application location XML file] - XML file that contains the locations to place the icon for the bundle application. For the XML file format, refer to IconLocation.xml located in /opt/novell/zenworks/share/zman/samples/bundles on a Linux server or *Installation_directory*:\Novell\Zenworks\share\zman\samples\ bundles on a Windows server.

-d, --distribution-schedule=[distribution schedule XML file] - XML file that contains the distribution schedule.

-l, --launch-schedule=[launch schedule XML file] - XML file that contains the launch schedule.

-a, --availability-schedule=[availability schedule XML file] - XML file that contains the availability schedule. For the schedule XML file templates, refer to the XML files located in /opt/novell/zenworks/share/zman/samples/schedules on a Linux server or *Installation_directory*:\Novell\Zenworks\share\zman\samples\ schedules on a Windows server.

-i, --install-immediately - Installs the bundle immediately after distribution.
  To use this option, you must also specify the distribution schedule. The distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-L, --launch-immediately - Launches the bundle immediately after installation.
  To use this option, you must also specify the distribution schedule. The distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-n, --distribute-now - Sets the distribution schedule to distribute the bundle immediately. If this option is specified, the --distribution-schedule and --distribute-on-device-refresh options are ignored. The --distribute-now, --distribute-on-device-refresh and --distribution-schedule

options are mutually exclusive and are used to set the distribution schedule. The --distribute-now option is considered first, followed by --distribute-on-device-refresh and --distribution-schedule.

-r, --distribute-on-device-refresh - Sets the distribution schedule to distribute the bundle on device refresh. If this option is specified, the --distribution-schedule option is ignored.

-s, --launch-on-device-refresh - Sets the launch schedule to launch the bundle on device refresh. If this option is specified, the --launch-schedule option is ignored.

-w, --wakeup-device-on-distribution - Wakes up the device by using Wake-On-LAN if it is shut down while distributing the bundle.

To use this option, you must also specify the Distribution schedule. The Distribution schedule can be specified by using the --distribution-schedule, --distribute-now, or --distribute-on-device-refresh option.

-B, --broadcast=[Broadcast address][...] - A comma-separated list of addresses used to broadcast the Wake-On-LAN magic packets. This option is used only if you choose to wake up the device by using Wake-On-LAN. A valid IP address is a valid value.

-S, --server=[Path of the Primary or Proxy Server objects relative to /Devices] [...] - A comma-separated list of Primary or Proxy Server objects used to wake up the device. This option is used only if you choose to wake up the device by using Wake-On-LAN.

-C, --retries=[Number of retries] - Number of times the Wake-On-LAN magic packets are sent to the device. This option is used only if you choose to wake up the device by using   Wake-On-LAN. The value must be between 0 and 5. The default value is 1.

-T, --timeout=[Time interval between retries] - The time interval between two retries. This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 2 and 10 (in minutes). The default value is 2.

workstation-add-policy (wap) (workstation object path) (policy or policy group) [...] [options]

Assigns policies to a workstation object.

Accepts the following options:

-c, --conflicts=[policy conflict resolution order] - Determines how policy conflicts are resolved. Valid values are userlast or 1, devicelast or 2, deviceonly or 3, useronly or 4. For userlast, device-associated policies are applied first, followed by user-associated policies. For devicelast, user-associated policies are applied first, followed by device-associated policies. For deviceonly, user-associated policies are ignored. For useronly, device-associated policies are ignored. If this option is not specified, userlast is taken as the default value.

-e, --enforce-now - Enforces the policy immediately on all assigned devices.

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

workstation-delete (wd) workstation object path) [...] [options]

Deletes one or more workstation objects.

(workstation object path) [...] - The path of the workstation objects (workstation, workstation folder or workstation group) relative to /Devices/Workstations. The wildcard * can be used in the object names if it is enclosed in quotations. Exercise caution while using wildcards for deleting objects.

Accepts the following options:

-r, --recursive - Deletes objects inside a folder recursively.

-f, --folder=[workstation folder] - The path of a workstation folder relative to `/Devices/Workstations`. If this option is specified, the path of the workstation objects is determined relative to this folder. This can be used to specify multiple workstation objects under the same folder.

workstation-folder-create (wfc) (folder name) [parent folder] [options]

Creates a new folder for containing workstations.

Accepts the following option:

--desc=[description] - Description for the folder.

workstation-group-add (wga) (workstation group path) (workstation path) [...] [options]

Adds workstations to a workstation group.

Accepts the following option:

-f, --folder=[workstation folder] - The path of a workstation folder relative to `/Devices/Workstations`. If this option is specified, the path of the workstation objects is determined relative to this folder. This can be used to specify multiple workstation objects under the same folder.

workstation-group-create (wgc) (group name) [parent folder] [options]

Creates a workstation group and adds members to it.

Accepts the following options:

--desc=[description] - Description for the group.

-m, --members=[workstation path][...] - The path of the workstations relative to `/Devices/Workstations`.

-f, --folder=[workstation folder] - The path of a workstation folder relative to `/Devices/Workstations`. If this option is specified, the path of the workstation objects is determined relative to this folder. This can be used to specify multiple workstation objects under the same folder.

workstation-group-members (wgm) [options] (workstation group path)

Lists members of a workstation group or a dynamic workstation group.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

workstation-group-remove (wgr) (workstation group path) (workstation path) [...] [options]

Removes workstations from a workstation group.

Accepts the following option:

-f, --folder=[workstation folder] - The path of a workstation folder relative to `/Devices/Workstations`. If this option is specified, the path of the workstation objects is determined relative to this folder. This can be used to specify multiple workstation objects under the same folder.

workstation-info (wi) (workstation path)

Lists detailed information about a workstation.

workstation-list (wl) [folder] [options]

Lists workstation objects.

Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, the contents of the subfolders are not listed.

-n, --namefilter=[filter string] - Displays results matching the specified name. The wildcards * and ? can be used if they are enclosed in quotation marks.

-t, --typefilter=[type filter][...] - Displays results matching the comma-separated list of object types specified. Valid values are `device`, `group`, and `folder`.

-c, --count - Displays the count of the results.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

-f, --flagfilter=[flag filter][...] - Displays results matching the comma-separated list of flags specified. Valid values are `retired`, `notretired`, `test`, and `non-test`.

workstation-list-bundles (wlb) (workstation object path) [options]

Lists bundles assigned to a workstation object.

Accepts the following options:

-a, --all - Lists both effective and non-effective bundles.

-e, --effective - Lists only effective bundles.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

workstation-list-groups (wlg) (workstation path) [options]

Lists groups of which the given workstation is a member.

Accepts the following option:

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

workstation-list-policies (wlp) (workstation object path) [options]

Lists policies assigned to a workstation object.

Accepts the following options:

-a, --all - Lists both effective and non-effective policies.

-e, --effective - Lists only effective policies.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

workstation-move (wmv) (workstation object path) [destination folder path]

Moves a workstation object to a different folder.

workstation-refresh (wrf) (workstation object path) [...] [options]

Refreshes the ZENworks Adaptive Agent in the workstations.

Accepts the following option:

-f, --folder=[workstation folder] - The path of a workstation folder relative to `/Devices/Workstations`. If this option is specified, the path of the workstation objects is determined relative to this folder. This can be used to specify multiple workstation objects under the same folder.

workstation-remove-bundle (wrb) (workstation object path) (bundle or bundle group) [...] [options]

Removes bundles assigned to a workstation object.

Accepts the following option:

-f, --folder=[bundle folder] - The path of a bundle folder relative to /Bundles. If this option is specified, the path of the bundle objects is determined relative to this folder. This can be used to specify multiple bundle objects under the same folder.

workstation-remove-policy (wrp) (workstation object path) (policy or policy group) [...] [options]

Removes policies assigned to a workstation object.

Accepts the following option:

-f, --folder=[policy folder] - The path of a policy folder relative to /Policies. If this option is specified, the path of the policy objects is determined relative to this folder. This can be used to specify multiple policy objects under the same folder.

workstation-rename (wr) (workstation object path) (new name)

Renames a workstation object.

workstation-reorder-bundles (wrob) (workstation object path) (current position) (new position)

Changes the order of bundles assigned to a workstation. Use the workstation-list-bundles command to get the order of assigned bundles.

workstation-reorder-policies (wrop) (workstation object path) (current position) (new position)

Changes the order of policies assigned to a workstation. Use the workstation-list-policies command to get the order of assigned policies.

workstation-retire (wret) (workstation object path)

Retires the selected workstation from your ZENworks system at its next refresh. Retiring a device is different from deleting a device. When you retire a device, its GUID is retained (as opposed to when you delete a device, which also deletes its GUID). As a result, all inventory information is retained and is accessible but all policy and bundle assignments are removed. If you reactivate the device in the future, its assignments are restored.

Accepts the following option:

-i, --immediate - Forces a device refresh to immediately retire the device.

workstation-set-as-test (wsat) (workstation object path)

Sets a workstation as a test device.

Examples

To set workstation1 as a test device:

zman wsat workstation1

workstation-set-as-nontest (wsan) (workstation object path)

Sets a workstation as a non-test device.

Examples

To set workstation1 as a non-test device:

zman wsan workstation1

workstation-unretire (wuret) (workstation object path)

Reactivates the selected workstation at its next refresh and reapplies all policy and bundle assignments that the device previously had.

Accepts the following option:

-i, --immediate - Forces a device refresh to immediately unretire the device.

workstation-view-all-bundles-status (wvabs) (workstation path) [options]

Displays the advanced deployment status of bundles assigned to a workstation.

Accepts the following options:

--statusfilter=[status type][...] - Filter on the status of Bundle Distribution and Install Events. Valid values are S, R, C, A, F, and P (Success, Partial Success, Completed, Partial Complete, Failure, and Pending). A comma-separated list of status types can be given.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

Examples

To view the deployment status of bundles assigned to workstation1:

`zman wvabs workstation1`

To view the deployment status of bundles assigned to workstation1 having bundle deployment status as success or pending:

`zman wvabs workstation1 --statusfilter S,P`

workstation-view-all-policies-status (wvaps) (workstation path) [options]

Displays the advanced deployment status of policies assigned to a workstation.

Accepts the following options:

--statusfilter=[status type][...] - Filters on the status of Policy Apply Event. Valid values are S, F, and P (Success, Failure, and Pending). A comma-separated list of status types can be given.

-s, --scrollsize=[scrollsize] - Number of results to be displayed at a time.

workstation-wakeup (ww) (workstation object path) [...] [options]

Wakes up a workstation by using Wake-On-LAN.

Accepts the following options:

-f, --folder=[workstation folder] - The path of a workstation folder relative to `/Devices/ Workstations`. If this option is specified, the path of the workstation objects is determined relative to this folder. This can be used to specify multiple workstation objects under the same folder.

-b, --broadcast=[Broadcast address][...] - A comma-separated list of addresses used to broadcast the Wake-On-LAN magic packets. This option is used only if you choose to wake up the device by using Wake-On-LAN. A valid IP address is a valid value.

-s, --server=[Path of the Primary or Proxy Server objects relative to `/Devices`][...] - A comma-separated list of Primary or Proxy Server objects used to wake up the device. This option is used only if you choose to wake up the device by using Wake-On-LAN.

-r, --retries=[Number of retries] - Number of times the Wake-On-LAN magic packets are sent to the device(s). This option is used only if you choose to wake up the device by using   Wake-On-LAN. The value must be between 0 and 5. The default value is 1.

-t, --timeout=[Time interval between retries] - The time interval between two retries. This option is used only if you choose to wake up the device by using Wake-On-LAN. The value must be between 2 and 10 (in minutes). The default value is 2.

ZENworks Server Commands

These commands are used to manage servers that have ZENworks 11 SP2 installed. ZENworks Server commands begin with `zenserver-` in the long form, or with the letters `zs` in the short form.

- "zenserver-backup (zsb) (file path)" on page 100
- "zenserver-restore (zsr) (file path)" on page 100
- "zenserver-retire (zsret) (server object path)" on page 100
- "zenserver-unretire (zsuret) (server object path)" on page 100

zenserver-backup (zsb) (file path)

Backs up the configuration files needed to duplicate this server on another ZENworks Server.

(file path) - Path to the file to which the ZENworks Server configuration files are backed up. If the server contains the embedded database, use the `database-backup` command to back up the database. To restore the server on another server installation, you must first restore the configuration files by using the `zenserver-restore` command, and then restore the database. The command prompts for a passphrase to encrypt the backed-up file. Make a note of the specified passphrase because you need to enter it for restoring the server.

zenserver-restore (zsr) (file path)

Restores the backed-up configuration files of the crashed ZENworks Server to a new ZENworks Server installation.

(file path) - Path to the file containing the backed-up configuration files to be restored. The server should have the same DNS name and the IP address as the server that is being restored. When the command prompts for a passphrase to decrypt the backed-up file, enter the passphrase that was used to encrypt the files when they were backed up.

zenserver-retire (zsret) (server object path)

Retires the selected ZENworks Primary Server from your ZENworks system at its next refresh. Retiring a device is different from deleting a device. When you retire a device, its GUID is retained (as opposed to when you delete a device, which also deletes its GUID). As a result, all inventory information is retained and is accessible, but all policy and bundle assignments are removed. If you unretire the device in the future, its assignments are restored.

Accepts the following option:

-i, --immediate - Forces a device refresh to immediately retire the device.

zenserver-unretire (zsuret) (server object path)

Reactivates the selected server at its next refresh to once again be a Primary Server and reapplies all policy and bundle assignments that the device previously had.

Accepts the following option:

-i, --immediate - Forces a device refresh to immediately unretire the device.

Global Options

The following options can be applied to any zman command:

- "-h, --help, ?, -?, /?" on page 101
- "--usage" on page 101
- "--terse" on page 101
- "--debug" on page 101

-
-
-
-
-
-
-
-
-

-h, --help, ?, -?, /?

Provides help at various levels. Refer to the Guide to Usage section for more details on using help.

--usage

Displays the syntax of the command.

--terse

Displays terse output.

--debug

Displays debugging output.

-V, --verbose

Enables verbose output.

--quiet

Quiets output, printing only error messages.

-U, --User=<username>

Provides a username. If it is not provided, you are prompted.

-P, --Password=<password>

Specifies a password. If it is not provided, you are prompted.

---host=<host>

Specifies the host name or IP address to connect to (default: localhost).

--port=<port>

Specifies the port that the server is listening on (default: 443).

--cleartext

Disables SSL for debugging purposes. If the port is not set to the clear text port, it connects to port 80 by default.

-R, --Redirect=<file path>

Redirects the output of the screen to a file. Use this option over the command line redirection operator (>) to save data in UTF-8 format and to preserve non-English characters.

-G, --ignore-global-file

Ignore the global options file specified in `zman-config.properties`.

Security

Running each command requires administrator authorization. To avoid entering administrator credentials for every command, use the `admin-store-credential` command to store the credentials. The credentials are encrypted and stored in the home folder of the operating system user running the command. The credentials only apply to that logged-on user. It is safer to clear the information by running the `admin-clear-credential` command after the execution of all the other commands is done.

IMPORTANT: Although command line options for entering usernames and passwords exist, avoid using them. In Linux, they display in the `ps` listing. In addition, pressing the Up-arrow or Down-arrow keys displays the previous commands entered in both Windows and Linux. Either enter the credentials when prompted or use the `admin-store-credential` command.

Files

`zman-config.properties` is a property file used to configure zman and is located in the following locations:

- **On Windows:** `%ZENWORKS_HOME%\conf\zman\properties`
- **On Linux:** `/etc/opt/novell/zenworks/zman/properties`

The various options available are:

- **RENDERER_SCROLLSIZE:** Controls the output of a list command globally. Results are retrieved and displayed in counts of scroll size specified. This setting can be overridden by using the --scrollsize option in the list commands.

- **GLOBAL_OPTIONS_FILE:** Global options can be added to every command by storing them in a file and specifying the path of the file as the value to this property. For Windows files, use \\ instead of \. The contents of the file should be in a single line similar to a command.

  For example:

  `GLOBAL_OPTIONS_FILE=C:\\zman\\globaloptions.txt`

  Sample content for the file is as follows:

  `--host 123.34.45.56 --port 8080 –cleartext --verbose`

  To bypass the options stored in this file, use the –ignore-global-file option. All the global options except, -U|--User and -P|--Password can be stored in the global options file. The `admin-store-credential` and `admin-clear-credential` commands should be used to add administrator credentials to each command. Refer to the Security section for guidelines on using administrator credentials.

- **GLOBAL_OPTIONS_HELP:** Global options for the commands are displayed in the help for every command. To turn off the display of global options, set `GLOBAL_OPTIONS_HELP=false`.

- **DEBUG_LEVEL:** Controls the volume of debug messages logged into the `zman.log` file located in `Installation_drive`:`\Novell\zenworks\logs` in Windows and `/var/opt/novell/log/zenworks` in Linux. Level 3, which is the default, logs the command entered and exception stack traces if any errors are encountered. Level 4 logs the command entered, the output of the command, and more debug messages.

Exit Codes

zman returns exit codes in the range of 0-255. 0 indicates successful execution of the command; 1-255 indicate an error in execution.

0 - Success.

Exit Codes for Parse Errors:

1 - An unknown parsing error has occurred.

2 - A mandatory argument is not specified.

3 - An unknown flag is specified

4 - A value is specified for a switch parameter.

5 - No value is specified for a flagged option.

6 - An unexpected additional argument is specified.

7 - An invalid value is specified.

8 - Unable to convert a value to the type expected.

9 - Unknown command.

10 - No options are entered when at least one is required. For example, to modify rights, you must specify at least the Assign option or the Revoke option.

Exit Codes for General Errors:

11 - An unknown error has occurred.

12 - The feature is not available because one of the dependent zman jar files is missing.

13 - Authentication failed.

14 - An empty string is specified for username or password, or the specified password does not meet the minimum length requirements.

15 - The ZENworks license has expired.

16 - Connection failure. The or the database device might be down.

17 - The ZENworks administrator does not have sufficient rights to perform this operation.

18 - Unable to read the certificate for establishing SSL communication.

19 - This command can be run only on a local host.

Exit Codes for General Object Related Errors:

21 - The object specified is not of the expected type. For example, a bundle is specified instead of a device.

22 - The object could not be found.

23 - An object with the same name already exists in the specified folder.

24 - The object specified is in a different directory than the specified folder.

25 - A parent folder cannot be moved into its subfolder.

26 - The object is not renamable.

27 - The object is not deletable.

28 - Attempting to delete a non-empty folder.

29 - The object is already a member of the group.

30 - The object is not member of the group.

31 - The object is already assigned to the device or user.

32 - The object is not yet assigned to the device or user.

33 - The object type is invalid to assign to a device or user.

34 - Assignment of the object to the device is not allowed. For example, the user-specific policy cannot be assigned to a device.

Exit Codes for Other General Errors:

41 - The specified name is invalid. For example, the object cannot have special characters such as * or % in its name.

42 - The absolute path starting with a front slash (/) is expected.

43 - An invalid schedule is specified.

44 - The XML file did not contain the expected input. For example, an exported action set XML file is given as input when an exported bundle XML file was expected.

45 - The XML file did not conform to the expected schema.

46 - An invalid GUID is specified.

47 - The specified value has exceeded the maximum length.

48 - A dependent argument has not been specified. For example, the command uses the option to install a bundle immediately on distribution, but does not specify the distribution schedule.

Exit Codes for File Related Errors:

61 - A directory is expected.

62 - The file does not exist or is a directory.

63 - The file already exists

64 - An error occurred while reading the file.

65 - An error occurred while writing to the file.

66 - An error occurred while creating the file.

67 - An error occurred while creating the directory

68 - An error occurred while copying the file.

69 - An error occurred while uploading the file.

70 - Unable to load the file because there is not sufficient disk space on the target device.

Exit Codes Specific to Registration Commands:

71 - The folder and the group being associated to the registration key or rule set are not of the same device type.

72 - A rule set should have at least one rule.

Exit Codes Common to Bundle and Policy Commands:

76 - Content creation failed.

77 - Copying the content file to the temporary directory was canceled because a file by that name already exists.

78 - Creation of a bundle or a policy with file content is being attempted on a server that is not a Content Server.

Exit Codes Specific to Bundle Commands:

81 - The file specified for the bundle icon is not a valid image file.

82 - The action set does not contain any actions to reorder, remove, or export.

83 - The action set does not contain any actions at the given position to reorder, remove, or export.

84 - The action set is not supported for the bundle type. For example, the Preboot action set is not supported for the Windows bundle.

85 - The action is not supported for the action set. For example, the Verify Bundle action is not supported for the Install action set.

86 - The action is non-deletable.

87 - The action type is singular; you cannot add the same action type more than once to an action set.

88 - The action type does not support dependent bundles. For example, you cannot specify a dependent bundle path for the Install MSI action.

89 - The specified dependent bundle path leads to a circular dependency of bundles.

90 - Failed to create the Wake-On-LAN schedule when assigning bundles.

91 - Failed to remove the Wake-On-LAN schedule when unassigning bundles.

Exit Codes Common to Device and User Command:

101 - The device or user does not have any bundles or policies associated to it.

Exit Codes Specific to User Commands:

106 - The User Source Certificate has expired.

107 - SSL is required to connect to the User Source.

Exit Codes Specific to License Commands:

116 - Unable to activate the license.

118 - The server does not have the Certificate Authority role.

119 - The server does not have the files required to enable the Certificate Authority role.

120 - The Certificate Authority used is not the ZENworks Internal Certificate Authority.

Exit Codes Specific to Administrator Command:

121 - The right does not belong to the rights category.

Exit Codes Specific to Database Commands:

126 - The database is not an embedded database.

127 - The SQL statement is invalid.

128 - An SQL exception has been encountered.

Exit Codes Specific to Deployment Commands:

131 - An invalid IP address or hostname has been specified.

132 - Credentials required to connect to the device are not specified.

133 - A managed device or a device that has been already scheduled for deployment is specified.

Exit Codes Specific to Report Commands:

136 - Failed to delete the report.

137 - Failed to submit a request for the report generation.

138 - Invalid report document.

Authors

Copyright 2005-2011, Novell, Inc (http://www.novell.com). All rights reserved.

zeninfocollect(1)

Name

zeninfocollect - A utility that lets you gather information to help you resolve problems that might arise when using Novell ZENworks 11 SP2.

Syntax

```
./zeninfocollect.py [options]
```

Description

By default, zeninfocollect gathers cache, server, client, configuration, hardware, and package data as well as log files. The information is packaged into a ZIP file on Windows and a tarball file on Linux, and placed in the location you specify.

By default, the zeninfocollect utility is not installed on any device.

To download and run the zeninfocollect utility on a Windows managed device:

1 Copy the `ZENworksInfocollect_Win.zip` file from the `ZENworks_Server\novell\zenworks\install\downloads\tools` directory, or download the file from ZENworks Control Center (*Configuration* tab > *Configuration Tasks* > *Download ZENworks Tools* > *Administrative Tools*).

2 Extract the `ZENworksInfocollect_Win.zip` file on the managed device.

3 At the command prompt of the managed device, go to the directory where the `.zip` contents are extracted and run `zeninfocollect.exe [options]`.

To download and run the zeninfocollect utility on a Linux managed device that has Python installed:

1 Copy the `ZENworksInfocollect_Linux.zip` file from the `ZENworks_Server`/novell/ zenworks/install/downloads/tools directory, or download the file from ZENworks Control Center (*Configuration* tab > *Configuration Tasks* > *Download ZENworks Tools* > *Administrative Tools*).

2 Extract the `ZENworksInfocollect_Linux.zip` file on the managed device.

3 At the command prompt of the managed device, go to directory where the `.zip` contents are extracted and run `./zeninfocollect.py [options]`.

List of Options

- "--output-dir=directory_name" on page 107
- "-h, --help" on page 107
- "-a, --no-cache" on page 107
- "-c, --no-client" on page 107
- "-d, --no-hardware" on page 107
- "-l, --no-logs" on page 107
- "-o, --no-config" on page 107
- "-p, --no-packages" on page 107
- "-s, --no-server" on page 107

`--output-dir=``directory_name`

Specify the directory name that must contain the output archive. If you do not specify a directory, the output is placed in user's temporary directory `%TEMP` on a Windows managed device, and `/tmp` on a Linux managed device.

`-h, --help`

Display the help information.

`-a, --no-cache`

Do not collect cache data.

`-c, --no-client`

Do not collect client data.

`-d, --no-hardware`

Do not collect hardware data.

`-l, --no-logs`

Do not collect logs.

`-o, --no-config`

Do not collect configuration data.

`-p, --no-packages`

Do not collect package data. Package data includes all version information for packages in the ZENworks 11 SP2 Configuration Management package repository.

`-s, --no-server`

Do not collect server data.

Authors

Copyright 2005-2011, Novell, Inc (http://www.novell.com). All rights reserved.

zac for Windows(1)

Name

zac - The command line management interface for the Novell ZENworks Adaptive Agent that is installed and running on Windows managed devices.

Syntax

```
zac command options
```

Description

The zac utility performs command line management functions on the ZENworks managed device, including installing and removing software bundles, applying policies, and registering and unregistering the device.

Guide to Usage

Most commands have a long form and a short form:

- Long form: `add-reg-key`
- Short form: `ark`

When both forms are available, the command is listed as follows:

```
add-reg-key (ark) arguments
```

When using the command, enter only the long form or the short form:

```
zac add-reg-key arguments
```

```
zac ark arguments
```

Arguments can be mandatory or optional. Mandatory arguments are included in angle brackets `<argument>`. Optional arguments are included in square brackets `[argument]`. If an argument includes a space, enclose it in quotation marks:

```
zac ark "arg 1"
```

Help Commands

- "/h or --help" on page 108

/h or --help

Displays information about the commands.

List of Commands

- "Authentication Satellite Server Commands" on page 109
- "Bundle Commands" on page 110
- "Certificate Commands" on page 111

Authentication Satellite Server Commands

- "authentication server reconfigure (asr) [-t all|config|jetty|casa] [-u username] [-p password]" on page 109
- "import-authentication-cert(iac)[-pk <private-key.der>] [-c <signed-server-certificate.der>] [-ca <signing-authority-public-certificate.der>] [-ks <keystore.jks>] [-ksp <keystore-pass-phrase>] [-a <signed-cert-alias>] [-ks <signed-cert-passphrase>] [-u username] [-p password] [-rc]" on page 109

```
authentication server reconfigure (asr) [-t all|config|jetty|casa] [-u username] [-
p password]
```

Reconfigures an enabled Authentication Satellite.

Examples:

To fetch the configuration files from the server:

```
zac asr -t config
```

To reconfigure the CASA signing certificate:

```
zac asr -t casa
```

To reconfigure the Jetty web server:

```
zac asr -t jetty -u Administrator -p password
```

To reconfigure the entire Satellite:

```
zac asr -t all -u Administrator -p password
```

If a username and password is required but is not provided on the command line you will be prompted.

```
import-authentication-cert(iac)[-pk <private-key.der>] [-c <signed-server-
certificate.der>] [-ca <signing-authority-public-certificate.der>] [-ks
<keystore.jks>] [-ksp    <keystore-pass-phrase>] [-a <signed-cert-alias>] [-ks
<signed-cert-passphrase>] [-u username] [-p password] [-rc]
```

Configures an Authentication Satellite device with externally signed certificates.

rc - Confirms reconfiguration of the Authentication Satellite Server so that the administrator is not prompted for reconfiguration.

Bundle Commands

bundle-install (bin) <bundle display name>

Installs the specified bundle. Use the `bundle-list` command to get a list of the available bundles and their display names.

Example:

`zac bin bundle1`

bundle-launch (bln) <bundle display name> [-noSelfHeal]

Launches the specified bundle. Use the `bundle-list` command to get a list of the available bundles and their display names.

Example to launch a bundle based on the display name:

`zac bln bundle1`

Example to launch a bundle based on the display name and turn selfhealing off if the launch action fails (by default, selfhealing is turned on):

`zac bln bundle1 -noSelfHeal`

bundle-list (bl)

Displays the list of bundles assigned to the device and the logged in user.

Example:

`zac bl`

bundle-props (bp) <bundle display name>

Displays the status, version, GUID, and requirements information for the specified bundle. Use the `bundle-list` command to get a list of the available bundles and their display names.

Example:

`zac bln bundle1`

bundle-refresh (br) <bundle display name or guid>

Refreshes information about the specified bundle.

Example:

`zac br bundle1`

bundle-uninstall (bu) <bundle display name>

Uninstalls the specified bundle. Use the `bundle-list` command to get a list of installed bundles and their display names.

Example:

`zac bu bundle1`

bundle-verify (bv) <bundle display name>

Verifies an installed bundle (specified by `bundle display name`) to ensure that no files have been removed or corrupted. Use the `bundle-list` command to get a list of the installed bundles and their display names.

Example:

`zac bv bundle1`

Certificate Commands

 ◆ "cert-info (ci) [ca certificate file path] [-u <username> -p <password>]" on page 111

cert-info (ci) [ca certificate file path] [-u <username> -p <password>]

Lists public key certificate information for each known ZENworks server or adds a trusted root certificate to the device trusted store. The file can be in ASN.1 DER format or base-64 encoded delimited by ----BEGIN CERTIFICATE---- and ----END CERTIFICATE--.

Example:

To list the certificate for each known ZENworks server:

`zac ci`

To add a trusted root certificate to the devices trusted store:

`zac ci c:\certs\mytrustcacert.der -u myuser -p mypassword`

Content Distribution Commands

 ◆ "cdp-checksum (cchk) [-l:<path to log>]" on page 111
 ◆ "cdp-verify-content (cvc) [-c] [-l:<path to log>]" on page 111
 ◆ "cdp-import-content (cic) <content path> [-l:<path to log>]" on page 111
 ◆ "wake-cdp (cdp) [replicate | cleanup]" on page 112

cdp-checksum (cchk) [-l:<path to log>]

Validates satellite content by computing the checksum on each file.

The optional log file details results of the checksum comparison.

Example:

`zac cchk -l:"C:\Program Files\Novell\ZENworks\logs\cchk.log"`

cdp-verify-content (cvc) [-c] [-l:<path to log>]

Compares the list of content IDs and their sync states on this CDP with what the Primary Servers thinks it should have.

You can use the following options:

c - Computes the checksum on the local content.

Example:

`zac cvc -l:"C:\Program Files\Novell\ZENworks\logs\cvc.log"`

cdp-import-content (cic) <content path> [-l:<path to log>]

Imports missing content from the directory specified by content-path, logging to the file specified by log-path.

Example:

```
zac cic c:\import_source_directory -l:"C:\Program
Files\Novell\ZENworks\logs\cic.log"
```

wake-cdp (cdp) [replicate | cleanup]

> Wakes the Content Distribution Point worker thread. You can use either of the following options:
>
> replicate - Downloads any new or changed content from the Content Distribution Point's parent ZENworks Server.
>
> cleanup - Removes any content that should no longer be stored on the Content Distribution Point.
>
> Examples:
>
> `zac cdp`
>
> `zac cdp replicate`
>
> This command is applicable only if the agent is promoted as a satellite.

Database Commands

- "statussender CleanUp" on page 112
- "statussender RollUp" on page 112

statussender CleanUp

> Runs the cleanup immediately and deletes entries in the MDStatus database that have not been updated for 14 days since the last successful rollup. By default, the cleanup is performed once a day and runs the first time the Agent Service is started. The cleanup method logs appropriate messages to the `zmd-messages.log` file when the log level is set to DEBUG.
>
> The MDStatus database is used for rolling bundle and policy status from the managed device to the ZENworks Server.

statussender RollUp

> Rolls up statuses to the MDStatus database that have been updated since the last time the status was rolled up successfully. By default, the status sender automatically rolls statuses up every 5 minutes.
>
> The MDStatus database is used for rolling bundle and policy status from the managed device to the ZENworks Server.

Imaging Commands

- "file-system-guid (fsg) [-d] [-r]" on page 112

file-system-guid (fsg) [-d] [-r]

> Displays, removes, or restores the workstation GUID in the file system in preparation for taking an image.
>
> For example:
>
> To display the GUID value:
>
> `zac fsg`
>
> To remove the GUID from the file system:
>
> `zac file-system-guid -d`
>
> To restore the GUID to the file system:

```
zac file-system-guid -r
```

Do not specify an option to print the the GUID value:

```
zac fsg
```

Inventory Commands

◆ "inventory (inv) [scannow | cdf]" on page 113

`inventory (inv) [scannow | cdf]`

Runs an inventory scan or opens the Collection Data Form.

Example to run an inventory scan:

```
zac inv scannow
```

Example to open the Collection Data Form:

```
zac inv cdf
```

Location Commands

◆ "config-location (cl)" on page 113

◆ "security-location (sl) [view | list | set <location>]" on page 113

`config-location (cl)`

Displays the configuration location. The configuration location determines which ZENworks server (or servers) the device connects to for authentication, configuration, content, and collection purposes.

Examples:

```
zac config-location
zac cl
```

`security-location (sl) [view | list | set <location>]`

Displays or changes the security location for the device. The security location determines which security policies (settings) are applied to the device.

Accepts the following optional arguments. If no argument is specified, the view argument is used.

view - Displays the current location.

list - Displays all defined locations.

set <location> - Changes to the specified location. <location> must be one of the defined locations.

Examples:

```
zac security-location view
zac sl
zac sl list
zac sl set office
```

Logging Commands

logger (log) [resetlog | level [MANAGED|ERROR|WARN|INFO|DEBUG] | managedlevel]

Changes or displays the logger configuration for the ZENworks Adaptive Agent.

You can use the following options:

resetlog - Resets the log.

level - If this option is used without a level, it displays the current managed logging level. If it is used with one of the levels, changes the logging level to the specified level.

managedlevel - Displays the Global Log level of the zone.

Example to reset the log file:

```
zac logger resetlog
```

Example to show the current log level:

```
zac logger level
```

Example to set the log level to DEBUG and above:

```
zac logger level DEBUG
```

Policy Commands

policy-list (pl)

Lists the policies that are currently being enforced on the device (effective policies). To list all policies (effective and non-effective), use the --all option.

Examples:

```
zac pl
zac pl --all
```

policy-refresh (pr)

Applies all of the policies assigned to the device and user.

Example:

```
zac pr
```

Registration Commands

add-reg-key (ark) <registration key>

Registers the device by using the specified key. Registration with keys is additive. If the device has previously been registered with a key and you register it with a new key, the device receives all group assignments associated with both keys.

Example:

`zac ark key12`

get-settings (gs) <registration key>

Lists the settings associated with the specified registration key.

Example:

`zac gs key1`

You can also use the `get-settings (gs)` command to list settings using various keys. For more information, see `get-settings (gs) <key>` in the Status Commands section.

register (reg) [-g] [-k <key>] [-u <username> -p <password] <ZENworks Server address:port>

Registers the device in a Management Zone.

You can use the following options:

g - Lets you create a new device object with a new GUID and password for the device if you have multiple devices with the same GUID. When you register a device by using this switch, all the associations (policies and bundles) assigned to the original device object are removed. You cannot use this option to create a new GUID for a Primary Server or a Satellite device. The local user must have Local Administrator rights to use this option.

k - Lets you register the device using the specified registration key.

p - Lets you specify the Management Zone administrator's password.

u - Lets you specify the Management Zone administrator's username.

Examples:

`zac reg -k key1 https://123.456.78.90`

`zac reg -k key1 -u administrator -p novell https://zenserver.novell.com:8080`

The port number is required only if the ZENworks Server is not using the standard HTTP port (80). If a username and password are not supplied, you are prompted for them.

reregister (rereg)[-u <username> -p <password>] <new guid>

Registers a device in the current zone and assigns it the GUID of an existing device object. The currently associated device object is deleted.

For example, if you image a device after replacing the hard drive, the device might get a new GUID. However, by using the reregister command, you can assign the device's GUID that it had before you replaced the hard drive.

Examples:

To reregister, specify a username and password:

`zac reregister -u myuser -p mypassword eaa6a76814d650439c648d597280d5d4`

To reregister and be prompted for a username and password:

`zac reregister eaa6a76814d650439c648d597280d5d4`

unregister (unr) [-f] [-u <username> -p <password>]

Removes the device's registration from the Management Zone.

Example:

```
zac unr -u administrator -p novell
```

The -f, -u, and -p parameters are optional. If you don't use the -u and -p parameters, you are prompted to enter a username and password. The -f parameter ignores the ZENworks database and forces the device to be unregistered locally; this option is necessary only if the device object has already been deleted from the ZENworks database or if the device cannot connect to the database.

reestablish-trust (retr) [-u <username> -p <password>]

Reestablishes trust with the current Management Zone. The username and password used must be of the Zone Administrator.

Example:

```
zac retr -u myuser -p mypassword
```

The -u and -p parameters are optional. If you don't use the -u and -p parameters, you are prompted to enter a username and password.

Status Commands

cache-clear (cc)

Clears the ZENworks cache on the device. This removes all entries in the cache database and deletes any cache files associated with those entries.

Example:

```
zac cc
```

NOTE: If your ZENworks administrator has enabled the self defense feature for the ZENworks Adaptive Agent, you must supply an override password before running the zac cc command. Otherwise, you receive the following message:

```
You do not have permission to clear the cache. Please contact your ZENworks
administrator.
```

You must request the override password from your ZENworks administrator. If he has not set an override password, he must do so before you can use the command. After you receive the password:

1. Double-click the ZENworks icon (z-icon) in the system tray, click *Agent* (under *Status*), then click the *Policy Override* link in the *Agent Security Settings* section to display the About box.

2. Click *Override Policy*, enter the override password, then click *Override*.

3. Go to a command line prompt and run the `zac cc` command.

4. After the cache is successfully cleared, return to the About box and click *Load Policy* to disable the password override.

dump-prop-pages (dpp) <target directory>

Outputs the HTML pages displayed in the ZENworks icon's property pages to files in the specified target directory.

Example:

`zac dpp c:\temp`

get-settings (gs) <key>

Lists the settings associated with the specified key.

Example:

`zac gs key1`

All valid ZENworks settings keys are stored in the `\Program Files\Novell\ZENworks\cache\zmd\settings` directory.

Example to list the Remote Management settings:

`zac gs RemoteManagement`

refresh (ref)[general | partial bundle <Bundle Display Name> [bypasscache]

Initiates a general refresh to refresh all bundles, policies, registration, and configuration settings; initiates a partial refresh to refresh all policies, registration, and configuration settings.

Use `bypasscache` to avoid using data from the server cache during the refresh. This option is useful for testing or troubleshooting.

Examples:

`zac ref general bypasscache`

`zac ref partial bypasscache`

set-proxy (sp) [options] <IP address/Hostname:port> [username] [password]

Specifies a proxy to contact rather than contacting a ZENworks Server directly. The options are:

/default - Sets a proxy that can be overriden by proxy settings from the Management Zone.
/clear - Clears the current proxy, but will use proxy settings from the Management Zone.

Examples:

`zac sp 123.456.78.90 administrator novell`

`zac sp /default 123.456.78.90`

If a username and password are not supplied, you are prompted for them.

winproxy-refresh (wpr)

Queries the Management Zone for proxy work assigned to this device.

Example:

`zac wpr`

zenhttp-status(zhs)

Lists port and tags for registered handlers.

Example:

```
zac zhs
```

This command is applicable only if the agent is promoted as a satellite.

`info-collect (zeninfo) [<targetfile>] [-q]`

Collects ZENworks support information, including cache data, configuration data, debug logs, product installation information, refresh times, status events, and basic system information. The information is packaged into a ZIP file and placed in the location you specify. If you do not specify a location, `${TEMP}\zeninfo-${DateTime}.zip` is used for Windows and `${TMPDIR}\zeninfo-${DateTime}.zip` is used for Linux. If you are experiencing problems with a managed device, Novell Support might ask you to run this command and send the resulting ZIP file to Novell to help troubleshoot your problem.

To run the zeninfo process in the background, run the following command:

```
zac zeninfo /tmp/zeninfo/ & echo $! > /tmp/zeninfo/zeninfo.pid
```

To stop the zeninfo process, run the following command:

```
kill `cat /tmp/zeninfo/zacinfo.pid`
```

You can use the following option:

q - Skip launching explorer after collection.

The `zeninfo` command can be run by the local administrators. If you are not a local administrator and you run the command, the system prompts you to enter the administrator credentials. You can also set the `AllowZenInfoWithoutAdminPwd` string value to `True`, which enables any user to run the `zeninfo` command. To set the `AllowZenInfoWithoutAdminPwd` string value, do the following:

1. Open the Registry Editor.

2. Go to `HKLM\Software\Novell\ZCM\`.

3. Set the `AllowZenInfoWithoutAdminPwd` string value to `True`.

WARNING: If the `AllowZenInfoWithoutAdminPwd` string value is set to `True`, the sensitive ZENworks Configuration Management settings and configuration information is visible also to the users who are not the local administrators.

`zone-config (zc) [-l]`

Displays information about the ZENworks Server that the device is accessing for configuration information (the Configuration server) or lists the information for the Configuration server.

Examples:

```
zac zc
zac zc -l
```

`statussender (sts) [options]`

This command is used to find the status information at the server. The options are:

Rollup - This is used to schedule an immediate rollup of the status information at the server inspite of the pre-scheduled time.

Cleanup - This is used to schedule an immediate cleanup of the status information into the MDStatus database.

zac for Linux(1)

Name

zac - The command line management interface for the Novell ZENworks Adaptive Agent that is installed and running on Linux managed devices.

Syntax

```
zac command options
```

Description

The zac utility performs command line management functions on the ZENworks managed device, including installing and removing software bundles, applying policies, and registering and unregistering the device.

Guide to Usage

This section contains a guide to general command formatting and conventions.

Most commands have a long form and a short form:

Long form: `add-reg-key`
Short form: `ark`

When both forms are available, the command is listed as follows:

```
add-reg-key (ark) options
```

When using the command, enter only the long form or the short form:

```
zac add-reg-key options
```

```
zac ark options
```

If an option includes a space, enclose it in quotation marks:

```
zac ark "option 1"
```

Mandatory and Optional arguments

> Arguments enclosed inside parentheses ( ) are mandatory and those enclosed by square brackets [ ] are optional.

Ellipsis (...)

> An ellipsis indicates that a command accepts multiple entries of the last argument type. The arguments must be separated by space or commas, depending on the command. For example:
>
> ```
> zman server-add-bundle [options] (server) (bundle) [...]
> ```
>
> The ellipsis indicates that `server-add-bundle` can accept multiple bundles. In the command level help, the multiple entry argument help is given as `[(bundle) (bundle) ... (bundle)]` for more clarity.

Help Commands

/? or --help

Displays help for the system.

COMMAND /?

Displays the description for a specified command.

/g or --group

Displays commands of the specified group.

List of Commands

Authentication Satellite Server Commands

authentication server reconfigure (asr) [-t all|config|jetty|casa] [-u username] [-p password]

Reconfigures an enabled Authentication Satellite.

Examples:

To fetch the configuration files from the server:

```
zac asr -t config
```

To reconfigure the CASA signing certificate:

```
zac asr -t casa
```

To reconfigure the Jetty web server:

```
zac asr -t jetty -u Administrator -p password
```

To reconfigure the entire Satellite:

```
zac asr -t all -u Administrator -p password
```

If a username and password is required but is not provided on the command line you will be prompted.

import-authentication-cert(iac)[-pk <private-key.der>] [-c <signed-server-certificate.der>] [-ca <signing-authority-public-certificate.der>] [-ks <keystore.jks>] [-ksp   <keystore-pass-phrase>] [-a <signed-cert-alias>] [-ks <signed-cert-passphrase>] [-u username] [-p password] [-rc]

Configures an Authentication Satellite device with externally signed certificates.

rc - Confirms reconfiguration of the Authentication Satellite Server so that the administrator is not prompted for reconfiguration.

Each Satellite must have its own individual server certificate and private key. For more information about generating external certificates, see the *ZENworks 11 SP2 Server Installation Guide*.

Bundle Commands

- "bundle-install (bin) [options] [bundlename | GUID | Pattern]" on page 121
- "bundle-launch (bln) [bundle display name] [-noSelfHeal]" on page 122
- "bundle-list (bl) [options]" on page 122
- "bundle-lock-add (bla) [options] [bundlename] [...]" on page 122
- "bundle-lock-list (bll)" on page 123
- "bundle-lock-remove (blr) [options] [bundleindex] [...]" on page 123
- "bundle-refresh (br) [bundle display name | guid]" on page 123
- "bundle-search (bse) [options] pattern" on page 123
- "bundle-uninstall (bu) [bundle display name | guid]" on page 124
- "bundle-verify (bv) [bundle display name]" on page 124

bundle-install (bin) [options] [bundlename | GUID | Pattern]

Installs the specified bundles based on the display name that match the specified regular expression or string, or the bundle GUID. Use the `bundle-list` command to get a list of the available bundles and their display names.

The following regular expressions are supported:

Regular Expression	Functionality
^abc	Matches string starting with 'abc'.
[abc]	Matches string containing 'a', 'b', or 'c'.
a*c	Matches zero or more times 'a' followed by 'c', 'ac','aac','aaac',or 'c'.

Accepts the following option:

-N, --dryrun - Tests and displays but does not actually perform the requested action. You can perform a dry run on a bundle version only once. This option is applicable only for the Install RPM actions of the bundle.

An example to install a bundle based on the bundle name:

`zac bin bundle1`

bundle-launch (bln) [bundle display name] [-noSelfHeal]

Launches the specified bundle. The actions in the Launch Action set are executed.

If you try to launch a bundle that is not installed, the bundle is first installed and then launched.

Use the `bundle-list` command to get a list of the available bundles and their display names.

An example to launch a bundle based on the display name:

`zac bln bundle1`

Example to launch a bundle based on the display name and to turn self-healing off if the launch action fails (by default, selfhealing is turned on):

`zac bln bundle1 -noSelfHeal`

bundle-list (bl) [options]

Displays the list of bundles assigned to the device and the logged-in user.

The output of this command is displayed in columns and provide the details of every bundle assigned or available for the managed device. The Display Name, Version, and Bundle Type columns represent the bundle display name, the version number of the bundle or sandbox if the bundle is a sandbox version, and the bundle type, respectively.

The Status column displays the following states:

Available: Displays this state if the bundle is installed.

Not Installed: Displays this state if the particular bundle version is not installed.

Downloaded: Displays this state if the contents are downloaded but the action in the Install Action set have not been executed.

This command accepts the following options:

-a, --all - Lists both the filtered and non-filtered bundles. Filtered bundles are chain bundles. By default, the non-filtered bundles are displayed.

-g, --show-guid - Displays the bundle GUID.

An example to list all the bundles:

`zac bl -a`

bundle-lock-add (bla) [options] [bundlename] […]

Locks the specified bundles that match the specified regular expression or string from being upgraded or removed.

The following regular expressions are supported:

Regular Expression	Functionality
^abc	Matches string starting with 'abc'.
[abc]	Matches string containing 'a', 'b', or 'c'.
a*c	Matches zero or more times 'a' followed by 'c', 'ac','aac','aaac',or 'c'.

Accepts the following options:

-li, --lock-install - Locks the bundle and prevents the installation of the bundle that is specified in the lock.

NOTE: Bundle locks are applicable only for Install action set.

-t, --type - Specify the bundle type to be locked. The available types are linux and patch. If no type is specified, the type, bundle, is selected.

bundle-lock-list (bll)

Lists the bundle locks. The bundle locks are sorted by ID number.

Example:

```
zac bll
```

bundle-lock-remove (blr) [options] [bundleindex] [...]

Removes the bundle lock that you specify by its number. You can find the number for each lock by using the `bundle-lock-list (bll)` command.

Accepts the following options:

-n - Removes the specified bundle locks by name. Ensure that the name does not contain any patterns.
--remove-all - Removes all bundle locks.

An example to remove a bundle lock abc:

```
zac blr -n abc
```

An example to remove all bundle locks:

```
zac blr --remove-all
```

bundle-refresh (br) [bundle display name | guid]

Refreshes information about the specified bundle.

Example:

```
zac br bundle1
```

bundle-search (bse) [options] pattern

Searches for bundles that match the specified regular expression or string.

The following regular expressions are supported:

Regular Expression	Functionality
a+bc	Matches one or more times 'a' followed by 'bc'.
	For example, matches one or more times 'a' followed by 'abc' or 'aabc'.
[a-z]*	Matches the empty string and strings with names containing only alphabets.
	For example, matches the empty string "", or strings with names such as acb and xyz with only alphabets.
a*bc	Matches zero or more times 'a' followed by 'bc'.
	For example, matches zero or more times 'a' followed by 'abc', 'aabc', 'aaabc', or 'bc'.

The output of this command is similar to the output of the `zac bl` command.

Accepts the following option:

-i, --installed-only - Searches only the list of installed bundles.

bundle-uninstall (bu) [bundle display name | guid]

Uninstalls the specified bundle. The actions that support the Undo action are executed while uninstalling the bundle. After the bundle uninstall is complete, the bundle state is displayed as Not Installed.

Use the `bundle-list` command to get a list of the installed bundles and their display names.

Example:

`zac bu bundle1`

bundle-verify (bv) [bundle display name]

Reinstalls the bundle and executes the Verify Action sets to ensure that no files have been removed or corrupted. Use the `bundle-list` command to get a list of the installed bundles and their display names.

Example:

`zac bv bundle1`

Certificate Commands

◆ "cert-info (ci) [ca certificate file path] [-u <username> -p <password>]" on page 124

cert-info (ci) [ca certificate file path] [-u <username> -p <password>]

Lists public key certificate information for each known ZENworks server or adds a trusted root certificate to the device trusted store. The file can be in ASN.1 DER format or base-64 encoded delimited by ----BEGIN CERTIFICATE---- and ----END CERTIFICATE--.

Example:

To list the certificate for each known ZENworks server:

`zac ci`

To add a trusted root certificate to the devices trusted store:

`zac ci c:\certs\mytrustcacert.der -u myuser -p mypassword`

Collection Rollup Commands

collection-clients (crc)

Displays devices that have uploaded files to this Collection.

Example:

```
zac crc
```

collection-queued (crq)

Displays collected files that have not yet been rolled up.

Example:

```
zac crq
```

collection-state (crs)

Displays the current state of the Satellite Collection role.

Example:

```
zac crs
```

collection-upload-orphans (cuo)

Finds orphaned files on the Satellite device and rolls them up to the parent collection server or deletes them if they have already been rolled up.

This command builds a list of the files in the folders under `/var/opt/novell/zenworks/work/collection` (except for the `collectionStatsDB` folder) and then tries to find the original upload information for each entry in the collection stats database.

If there is an entry for a file in the database, and it shows that the file has not been rolled up, it rolls the file up. If the entry shows that the file has already been rolled up, it deletes the file on the Satellite device. If there is no entry for a file in the database, the file is rolled up. This command also lists any files that were not uploaded or deleted.

Before running this command, you should run the `zac crw` command to send any pending files to the parent server.

Example:

```
zac cuo
```

collection-wake (crw)

Sends collected files to the parent server.

Example:

```
zac crw
```

Content Distribution Commands

cdp-checksum (cchk) [-l:<path to log>]

Validates satellite content by computing the checksum on each file.

The optional log file details results of the checksum comparison.

Example:

```
zac cchk -l:"/var/opt/novell/zenworks/logs/cchk.log"
```

cdp-verify-content (cvc) [-c] [-l:<path to log>]

Compares the list of content IDs and their sync states on this CDP with what the Primary Servers thinks it should have.

You can use the following options:

c - Computes the checksum on the local content.

Example:

```
zac cvc -l:"/var/opt/novell/zenworks/logs/cvc.log"
```

cdp-import-content (cic) <content path> [<path to log>]

Imports missing content from the directory specified by content-path, logging to the file specified by log-path.

Example:

```
zac cic /usr/content_Dir /var/opt/novell/zenworks/logs/cic.log
```

distribution-info (cdi) [option]

Displays statistical information about the Content role. You can use the following options:

all - Executes all options marked with *.
ss - Shows sync status *.
rs - Gets the Content Repository size *.
da - Lists devices that have retrieved data from this device *.
ne - Gets the number of content distribution errors *.
ds - Gets the size of data served by this device *.
fs - Gets the number of files served *.
ah [n] - Displays *n* (default all) entries from the access history.
ch - Clears the history.

eh *<file>* - Exports the access history to *<file>*, where the path is included and the file itself must not exist.

Examples:

```
zac cdi
zac cdi rs
```

distribution-state (cds)

Displays the state of the Satellite Content role. Displays if the role is enabled/disabled, the status (Starting, Active, Stopping, Stopped), the Parent URLs, and the replication schedule.

Example:

```
zac cds
```

distribution-wake (cdp) [replicate | cleanup]

Wakes the Content Distribution Point worker thread. You can use either of the following options:

replicate - Downloads any new or changed content from the Content Distribution Point's parent ZENworks Server.

cleanup - Removes any content that should no longer be stored on the Content Distribution Point.

Examples:

```
zac cdp
```

```
zac cdp replicate
```

Core Commands

get-settings (gs) <registration key>

Lists the effective settings associated with the specified registration key.

Example:

```
zac gs key1
```

Example to list the Remote Management settings:

```
zac gs RemoteManagement
```

All the valid ZENworks settings keys are stored in the `/var/opt/novell/zenworks/zmd/cache/settings/` directory.

get-pref (get) <preference name>

Displays the preferences that are currently set.

Example to display all preferences that are set:

```
zac get-pref
```

Example to display all preferences that have names beginning with abc:

```
zac get-pref abc
```

set-pref (set) <preference name> <preference value>

Sets the value for a preference.

Example to set the value of preference abc to 123:

```
zac set-pref abc 123
```

NOTE: If you want to configure a proxy server for a managed device, ensure that the managed device has both the `proxy-host` and `proxy-port` preferences configured. If the proxy server requires authentication, the managed device must additionally have the `proxy-username` and `proxy-password` preferences configured.

External Service Commands

These commands are used to manage external services.

external-service-add (esa) --type (type) (url) (Service Name) [Properties]

Adds external package repositories like YUM, ZYPP, or the Local directory that contains the RPMs. All the type-specific properties should start with -D.

You can list the service types that you can add by using the `zac est` command.

It supports the following types:

- YUM: Adds repository of type YUM. Accepts the following properties:

  -Dusername=value -Dpassword=value - This property is optional and is used while adding an authenticated YUM repository.
  -DsyncExternal=true - This property is optional and is used to synchronize with the Local Package Management tool.

  For example, run the following command to add a YUM service type:

```
zac esa --type YUM -Dusername=username -Dpassword=passwd https://nu.novell.com/
sles11 sles11-yum-repo
```

- ZYPP: Adds repository of type ZYPP. Accepts the following properties:

  -DsyncExternal=true - This property is optional and is used to synchronize with the Local Package Management tool.

  For example, run the following command to add a ZYPP service type:

```
zac esa --type ZYPP http://zypp.source.com/sles11/x86_64 sles11-zypp-repo
```

- Mount: Adds a repository of type Mount. A Mount type adds a service from the Local directory that contains the RPMs. Accepts the following properties:

  -Drecursive=(Value) - This property is optional. The valid values are true or false. If true, the RPMs are added recursively from all the directories under the specified absolute path.
  -DsyncExternal=true - This property is optional and is used to synchronize with the Local Package Management tool.

  For example, run the following command to add packages recursively from all directories by using the Mount service type:

```
zac esa --type mount -Drecursive=true file:///mnt/sles11-cd/i586 sles11-rpms
```

NOTE: The ZYPP or Mount services do not synchronize with the YUM Package Management Tool on Red Hat devices.

external-service-delete (esd) (index | name | url | --all)

Removes the specified external services.

The `--all` option deletes all the external services.

external-service-list (esl)

Lists all the external services.

The Source column displays the ZENworks Agents that were added by using the `zac` command and displays the ZENworks Server that was added through an External Service Policy.

The Status column displays whether the external service is active or inactive. You cannot refresh the external services that display their status as Inactive.

The State column displays whether the external service is enabled or disabled.

external-service-refresh (esr) (index | name | url | --all)

Refreshes the specified external services.

The `--all` option refreshes all the services.

external-service-state (ess) (enable | disable) (index | name | url | --all)

Enables or disables the specified external services.

The `--all` option changes the service state of all the services.

external-service-types (est)

Lists the available service types.

Imaging Commands

◆

file-system-guid (fsg) [-d] [-r]

Removes and restores the workstation GUID in the file system in preparation for taking an image.

Example to remove the GUID from the file system:

zac file-system-guid -d

Example to restore the GUID to the file system:

`zac file-system-guid -r`

Example to print the GUID value:

`zac fsg`

Inventory Commands

◆

inventory (inv) [scannow | cdf]

Runs an inventory scan or opens the Collection Data Form.

Example to run an inventory scan:

`zac inv scannow`

Example to open the Collection Data Form:

`zac inv cdf`

Local User Management Commands

These commands are used to manage local users and their privileges.

user-add (ua) (username)

Adds a local user to the ZENworks privilege store.

Example:

```
zac ua username
```

user-delete (ud) (username)

Deletes a local user from the ZENworks privilege store.

Example:

```
zac ud username
```

user-list (ul)

Lists the following:

- User names and privileges for the local users added to the ZENworks privilege store.
- Privileges for the local users not added to the ZENworks privilege store (default user). The default user is represented by * (asterisk).

Example:

```
zac ul
```

user-edit (ue) (username) [+/-Privilege]

Edits the local user privileges. You use + (plus) to add a new privilege and - (minus) to remove an existing privilege.

By default, a newly created local user has the View privilege. The local users that are not added to the ZENworks privilege store inherit the privilege assigned to the default user that is represented by * (asterisk). You can change the privileges assigned to any local user and assign the user the following privileges:

Privilege	Functionality
View	View or list the information on bundles, policies, configuration, and so on.
Install	Install or uninstall bundles, policies, packages, and so on.
Configure	Run the configuration commands.
SuperUser	Execute all the zac commands because a SuperUser is assigned all the privileges. By default, the root user is assigned the SuperUser privilege, which cannot be modified.

Examples:

If you want to allow the local users in the ZENworks privilege store to view, install, and uninstall bundles and policies, you can assign the `View` and `Install` privileges to these users by running the following command:

```
zac ue username +Install
```

If you want to allow the local users that are not added to the ZENworks privilege store to view, install, and uninstall bundles and policies, you can assign the `View` and `Install` privileges to to the `*` user by running the following command:

```
zac ue '*' +Install
```

user-privilege-export (upe) (.xml filename)

Exports a list of local users and their privileges to a specified `.xml` file for reusing this list on other managed devices.

Example:

```
zac upe /tmp/privileges.xml
```

After exporting the list, you can import it to other managed devices by running the `user-privilege-import (upi)` command.

user-privilege-import (upi) (.xml filename)

Imports a list of local users and their privileges to other managed devices from a specified `.xml` file. This list can be exported to the `.xml` file by running the `user-privilege-export (upe)` command.

Example:

```
zac upi /tmp/privileges.xml
```

Location Commands

-

config-location (cl)

Displays the configuration location. The configuration location determines which ZENworks server (or servers) the device connects to for authentication, configuration, content, and collection purposes.

Examples:

```
zac config-location
```

```
zac cl
```

Logging Commands

-

logger [resetlog | level [MANAGED|ERROR|WARN|INFO|DEBUG] | managedlevel]

Changes the logger configuration.

You can use the following options:

resetlog - Resets the log.

level - If used without a level, displays the current managed logging level. If used with one of the levels, changes the logging level to the specified level.

managedlevel - Displays the Global Log level of the zone.

Example to reset the log file:

```
zac logger resetlog
```

Example to show the current log level:

```
zac logger level
```

Example to set the log level to DEBUG and above:

```
zac logger level DEBUG MANAGED
```

Package Management Commands

- "checkpoint-add (cpa) (name) [date]" on page 132
- "checkpoints-list" on page 132
- "checkpoint-remove (cpr) (name) [...]" on page 132
- "dist-upgrade (dup) [options] [bundle name|service name] [...]" on page 132
- "info (info) [option] pattern" on page 133
- "install (in) [options] (package name) [...]" on page 133
- "list-updates (lu) [options] [bundle name|service name] [...]" on page 134
- "lock-add (la) (name or pattern)[version]" on page 134
- "lock-delete (ld) (lock number) [...]" on page 134
- "lock-list (ll)" on page 134
- "packagehistory (ph) [options] [...]" on page 135
- "remove (rm) [options] (package name) [...]" on page 135
- "rollback (ro) [options] [date or check point]" on page 135
- "search (se) [options] [pattern]" on page 136
- "update (up) [options] [bundle name|service name] [...]" on page 136
- "what-provides (wp) pattern" on page 136
- "what-requires (wr) pattern" on page 136

checkpoint-add (cpa) (name) [date]

Adds a checkpoint. If the date is not specified, adds a check point with the current date.

checkpoints-list

Lists the saved checkpoints.

checkpoint-remove (cpr) (name) [...]

Removes the specified checkpoints.

dist-upgrade (dup) [options] [bundle name|service name] [...]

Performs a distribution upgrade. This command performs an update of all packages with a special resolver algorithm which takes care of package architecture change, package splits, pattern and product updates, etc. If no arguments are provided, packages from all ZENworks bundles and external services are considered for distribution upgrade.

Accepts the following options:

-d, --download-only - Only downloads packages.
-N, --dry-run - Tests and displays the requested actions but does not actually perform them.

-y, --no-confirm - Does not prompt for confirmation.

For example, you can upgrade the distribution from SLES 10 SP2 to SLES 10 SP3.

`info (info) [option] pattern`

Displays package information for the specified package that matches the specified pattern.

The following regular expressions are supported:

Regular Expression	Functionality
^abc	Matches string starting with 'abc'.
[abc]	Matches string containing 'a', 'b', or 'c'.
a*c	Matches zero or more times 'a' followed by 'c', 'ac','aac','aaac',or 'c'.

The command displays the Name, Version, Architecture, and Description of the packages.

Accepts the following options:

--provides - Lists packages and files that is provided by the package that matches the specified pattern.

--requires - Lists packages and files that is required by the package that matches the specified pattern.

`install (in) [options] (package name) […]`

Installs the specified packages. If you specify a package with a version, the exact version of the package is installed. If you specify a package with a name, the best version of the package is installed.

NOTE: If you want to downgrade a package, specify the version to which you want to downgrade the package.

Accepts the following options:

-b, --bundle [bundle name] - Installs a package from the specified bundle. The bundle that you specify can also be a Linux Dependency Bundle. However, to install a Linux Dependency Bundle, the Publish Packages should be set to True.

-d, --download-only - Downloads packages to the `zen` cache directory. By default, the zen cache directory is located in `/var/opt/novell/zenworks/zmd/cache`.

-N, --dry-run - Tests and displays the requested actions but does not actually perform them.

-y, --no-confirm - Does not prompt for confirmation.

-f, --freshen - Installs packages only if an earlier version of the package exists.

-s, --service [service name] - Installs a package from the specified service.

-p, --pattern [pattern name] - Installs a package of a specified pattern. However, only the mandatory packages part of the pattern is installed.

--entire-bundle [bundle name] […] - Installs all packages from the specified bundles. You can specify a single bundle or multiple bundles.

--ignore-dependency - Does not check for dependency. The dependant packages are skipped and only the packages are installed.

-r, --regex-pattern <regex pattern> [<regex pattern>]- Installs all the packages that match the specified regular expression.

The following regular expressions are supported:

Regular Expression	Functionality
^abc	Matches string starting with 'abc'.
[abc]	Matches string containing 'a', 'b', or 'c'.
a*c	Matches zero or more times 'a' followed by 'c', 'ac','aac','aaac',or 'c'.

Additionally, the command also lists the dependent packages to successfully complete the installation.

list-updates (lu) [options] [bundle name│service name] […]

Displays available updates in the specified bundles and services. You can specify a single bundle or service, or multiple bundles or services for which you want to display the updates. If no bundle or service is specified, the command displays all the available updates.

Accepts the following option:

-t, --type [type] - Lists updates from the specified service type.

lock-add (la) (name or pattern)[version]

Adds a package lock rule. This command locks the specified packages, which if match the specified regular expression or string, are not upgraded or removed. The package name can be specified individually, with wildcard patterns, or even with version number relations. The version number should be specified within single or double quotes.

Relational Operator	Functionality
=	Locks only the specific package version.
<	Locks all versions of the package older than the specified version, excluding the specified version.
>	Locks all versions of the package later than the specified version, excluding the specified version.
<=	Locks all versions of the package older than the specified version, as well as the specified version.
>=	Locks all versions of the package later than the specified version, as well as the specified version.

If you want to install a specific version of the package, this command first checks if the package version has been locked, then installs the package version only if it is not locked. For example, assume that all the later versions of the package "X 1.7" have been locked by using the zac la X '>' 1.7 command. If you try to install the X 1.9 package by using the zac in X (1.9) command, the installation fails.

lock-delete (ld) (lock number) […]

Deletes the package lock that you specify by its number. You can find the number for each lock with the lock-list (ll) command.

lock-list (ll)

Lists the package locks.

packagehistory (ph) [options] […]

Lists package transaction history that is performed by using ZENworks. This command displays the package name, date, operation, version, and release.

Accepts the following option flags:

-n, --search-name [package name] - Searches by package name

-a, --search-action [action name] - Searches by action name. The valid action names are None, Install, Removal, and Upgrade.

-d, --days-back [days] - Maximum number of days to look back (default 30).

-c, --clean-history - Cleans package history.

-o, --search-operation [operation] - Searches based on the operation performed. The operation performed can be Install, Removal, or Upgrade.

remove (rm) [options] (package name) […]

Removes the specified packages.

Accepts the following options:

--entire-bundle [bundle name] - Removes all the packages from the specified bundle.

-N, --dry-run - Tests and displays but does not actually perform the requested actions.

-y, --no-confirm - Does not prompt for confirmation.

-r, --regex-pattern <regex pattern> [<regex pattern>] - Removes all the packages matching the specified regular expression.

The following regular expressions are supported:

Regular Expression	Functionality
^abc	Matches string starting with 'abc'.
[abc]	Matches string containing 'a', 'b', or 'c'.
a*c	Matches zero or more times 'a' followed by 'c', 'ac','aac','aaac',or 'c'.

Additionally, this command also lists the dependent packages to successfully complete the removal of the specified package.

rollback (ro) [options] [date or check point]

Rolls back package transactions to the time and date you specify, if the rollback preference is set. By default, the rollback preference is disabled. You can enable the rollback preference by running the zac set rollback true command.

If you want to roll back to a particular version of the package, that version of the package and its dependencies should be available as part of the service or the assigned bundle. If the version of the package or its dependencies is not available, the rollback fails.

Additionally, this command also lists the dependent packages that are required to successfully roll back.

Accepts the following options:

-d, --download-only - Only downloads packages.

-N, --dry-run - Tests and displays but does not actually perform the requested actions.

-y, --no-confirm - Does not prompt for confirmation.

-p, --package (package name) - Name of the package that needs to be rolled back.

search (se) [options] [pattern]

Searches for packages that match the specified pattern. If no arguments are provided, it displays all the packages. This command displays the source name, source type, package name, version, and architecture.

Accepts the following options:

-b, --bundle [bundle name] - Searches packages in the specified bundle.

-d, --search-in-description - Searches in package descriptions but not package names.

-i, --installed-only - Shows only installed packages.

-u, --uninstalled-only - Shows only uninstalled packages.

-s, --service [service name] - Searches packages in the specified service.

-t, --type [service type] - Searches packages, based on the specified service type.

-p, --pattern [pattern name] - Shows the available patterns.

--sort-by-bundle - Sort packages by bundle, not by name. Default sorting is by package name.

--show-nevra - Shows the NEVRA details of the packages.

update (up) [options] [bundle name|service name] […]

Updates the installed packages. You can specify a single bundle or service name, or multiple bundles or service names, for which to update the packages. If no bundle or service name is specified, all the packages are updated.

If there are installed packages that have the same name and architecture but different versions, only packages with the higher version are upgraded.

Accepts the following options:

-d, --download-only - Only downloads packages.

-N, --dry-run - Tests and displays but does not actually perform the requested actions.

-y, ---no-confirm - Does not prompt for confirmation.

-t, --type [service-type] - Updates packages from the specified external service type.

what-provides (wp) pattern

Displays packages that provide the library, program, or package that matches the specified pattern. Only the * wildcard pattern is supported.

Example to display the packages ending with a:

zac wp *a

what-requires (wr) pattern

Lists packages that require the library, program, or package that matches the specified pattern.

An example to display packages starting with a:

zac wr a*

Policy Commands

- "policy-list (pl)" on page 136
- "policy-refresh (pr)" on page 137

policy-list (pl)

Lists the policies that are currently being enforced on the device (effective policies). To list all policies (effective and non-effective), use the --all option.

Examples:

```
zac pl
zac pl --all
```

policy-refresh (pr)

Applies all of the policies assigned to the device and user.

Example:

```
zac pr
```

Registration Commands

add-reg-key (ark) <registration key>

Registers the device by using the specified key. Registration with keys is additive. If the device has previously been registered with a key and you register it with a new key, the device receives all group assignments associated with both keys.

Example:

```
zac ark key12
```

register (reg) [-k <key>] [-u <username> -p <password] <ZENworks Server address:port>

Registers the device in a Management Zone.

Examples:

```
zac reg -k key1 https://123.456.78.90
zac reg -k key1 -u administrator -p novell https://zenserver.novell.com:8080
```

The port number is required only if the ZENworks Server is not using the standard HTTP port (80). If a username and password are not supplied, you are prompted for them.

reregister (rereg)[-u <username> -p <password>] <new guid>

Registers a device in the current zone and assigns it the GUID of an existing device object. The currently associated device object is deleted.

For example, if you image a device after replacing the hard drive, the device might get a new GUID. However, by using the reregister command, you can assign the same GUID that it had before you replaced the hard drive.

Examples:

To reregister, specify a username and password:

```
zac reregister -u myuser -p mypassword eaa6a76814d650439c648d597280d5d4
```

To reregister and be prompted for a username and password:

```
zac reregister eaa6a76814d650439c648d597280d5d4
```

unregister (unr) [-f] [-u <username> -p <password>]

Removes the device's registration from the Management Zone.

Example:

```
zac unr -u administrator -p novell
```

The -f, -u, and -p parameters are optional. If you don't use the -u and -p parameters, you are prompted to enter a username and password. The -f parameter ignores the ZENworks database and forces the device to be unregistered locally; this option is necessary only if the device object has already been deleted from the ZENworks database or if the device cannot connect to the database.

reestablish-trust (retr) [-u <username> -p <password>]

Reestablishes trust with the current Management Zone. The username and password used must be of the Zone Administrator.

Example:

```
zac retr -u myuser -p mypassword
```

The -u and -p parameters are optional. If you don't use the -u and -p parameters, you are prompted to enter a username and password.

Satellite Role Commands

- "remint-satellite-cert (rsc)" on page 138
- "satellite-cachecfg (satcfg)" on page 138
- "satellite-clearcache (satcc)" on page 138
- "satellite-refreshcfg (satrc)" on page 138
- "satellite-roles (satr)" on page 138
- "imgsatserver status (iss status)" on page 139

remint-satellite-cert (rsc)

Remints the SSL certificate for Satellite.

Example: `zac rsc`

satellite-cachecfg (satcfg)

Reads Satellite roles from cache.

Example:

```
zac satcfg
```

satellite-clearcache (satcc)

Clears Satellite configuration information from cache.

Example:

```
zac satcc
```

satellite-refreshcfg (satrc)

Refreshes Satellite roles.

Example:

```
zac satrc
```

satellite-roles (satr)

Reports on current Satellite roles and their state.

Example:

```
zac satr
```

imgsatserver status (iss status)

Displays the device's imaging role properties, including its current status, PXE service status, image files, and imaging statistics.

Current Status: Displays whether your device is currently configured to perform Imaging operations. The current status is *Active* when the device is promoted to a satellite with the Imaging role.

PXE Service Status: Displays whether the Proxy DHCP service is enabled on the device.

PXE Requests: The number of imaging requests of any kind that have been received by the Imaging Server since it was last started. This includes requests that failed, were denied, or were referred to other Imaging Servers. Information about each of these requests, such as the source, type, date/time, and results, is logged on the Imaging Server.

Images Sent: The number of images that the Imaging Server has sent to imaging clients since the Imaging Server was last started. This includes only images that were retrieved from this Imaging Server.

Images Received: The number of new images that have been received and stored on the Imaging Server since it was last started. This includes images that were received through client referrals.

Example:

```
zac iss status
```

Status Commands

- "agent-properties (agp)" on page 139
- "cache-clear (cc)" on page 140
- "refresh (ref)[general [bypasscache] | partial [bypasscache] | bypasscache]" on page 140
- "zenhttp-status(zhs)" on page 140
- "zone-config (zc)" on page 140
- "get-settings (gs) <key>" on page 140

agent-properties (agp)

Provides information about the ZENworks Adaptive Agent. The following fields provide information about the ZENworks Adaptive Agent:

Device Address. The IP address of your device.

Device Name. The computer name for your device.

Device State. The state of the device - Managed, Unmanaged, Retired, or Unknown. The Unknown state is displayed only if there is an error.

Last Contact With Server. The last time the Adaptive Agent contacted the ZENworks Server.

Next Contact With Server. The time scheduled for the Adaptive Agent to contact the ZENworks Server.

Primary User. The currently logged in user.

Example:

```
zac agp
zac agent-properties
```

cache-clear (cc)

Clears the ZENworks cache on the device. This removes all entries in the cache database and deletes any cache files associated with those entries.

Example:

`zac cc`

refresh (ref)[general [bypasscache] | partial [bypasscache] | bypasscache]

Initiates a general refresh to refresh all registration and configuration settings; or initiates a partial refresh to refresh all registration and configuration settings.

Use `bypasscache` to avoid using data from the server cache during the refresh. This option is useful for testing or troubleshooting.

Examples:

`zac ref general bypasscache`

`zac ref partial bypasscache`

`zac ref bypasscache`

zenhttp-status(zhs)

Lists port and tags for registered handlers.

Example:

`zac zhs`

zone-config (zc)

Displays information about the ZENworks Server that the device is accessing for configuration information (the Configuration server) or lists the information for the Configuration server.

Examples:

`zac zc`

get-settings (gs) <key>

Lists the settings associated with the specified key.

Example:

`zac gs key1`

All valid ZENworks settings keys are stored in the `/var/opt/novell/zenworks/cache/zmd/settings` directory.

Example to list the Remote Management settings:

`zac gs RemoteManagement`

zac for Macintosh(1)

Name

zac - The command line management interface for the Novell ZENworks Adaptive Agent that is installed and running on Macintosh managed devices.

Syntax

```
zac command options
```

Description

The zac utility performs command line management functions on the ZENworks managed device, including installing and removing software bundles, applying policies, and registering and unregistering the device.

Guide to Usage

This section contains a guide to general command formatting and conventions.

Most commands have a long form and a short form:

Long form: `add-reg-key`
Short form: `ark`

When both forms are available, the command is listed as follows:

```
add-reg-key (ark) options
```

When using the command, enter only the long form or the short form:

```
zac add-reg-key options
```

```
zac ark options
```

If an option includes a space, enclose it in quotation marks:

```
zac ark "option 1"
```

Mandatory and Optional arguments

> Arguments enclosed inside parentheses ( ) are mandatory and those enclosed by square brackets [ ] are optional.

Ellipsis (...)

> An ellipsis indicates that a command accepts multiple entries of the last argument type. The arguments must be separated by space or commas, depending on the command. For example:
>
> ```
> zman server-add-bundle [options] (server) (bundle) [...]
> ```
>
> The ellipsis indicates that `server-add-bundle` can accept multiple bundles. In the command level help, the multiple entry argument help is given as `[(bundle) (bundle) ... (bundle)]` for more clarity.

List of Commands

Bundle Commands

bundle-install (bin)

Installs the specified bundles based on the display name that match the specified regular expression or string, or the bundle GUID.

Use the `bundle-list` command to get a list of the available bundles and their display names.

An example to install a bundle:

`zac bin MacBundle`

bundle-launch (bln)

Launches the specified bundle. The actions in the Launch Action set are executed. If you try to launch a bundle that is not installed, the bundle is first installed and then launched.

Use the `bundle-list` command to get a list of the available bundles and their display names.

An example to launch a bundle based on the display name:

`zac bln MacBundle`

bundle-list (bl)

Displays the list of bundles assigned to the device and the logged-in user.

The output of this command is displayed in columns and provide the details of every bundle assigned or available for the managed device. The Display Name, Version, and Bundle Type columns represent the bundle display name, the version number of the bundle or sandbox if the bundle is a sandbox version, and the bundle type, respectively.

The Status column displays the following states:

Available: Displays this state if the bundle is installed.

Not Installed: Displays this state if the particular bundle version is not installed.

Downloaded: Displays this state if the contents are downloaded but the action in the Install

Action set have not been executed.

An example to list all the bundles:

```
zac bl
```

bundle-lock-add (bla) [options] [bundlename] [...]

Locks the specified bundles that match the specified regular expression or string from being upgraded or removed.

Bundle locks are applicable only for Install action set.

Accepts the following options:

-li, --lock-install - Locks the bundle and prevents the installation of the bundle that is specified in the lock.

-t, --type - Specify the bundle type to be locked. The available types are mac and patch. If no type is specified, the type, bundle, is selected.

bundle-lock-list (bll)

Lists the bundle locks. The bundle locks are sorted by ID number.

An example to list the locked bundles:

```
zac bll
```

bundle-lock-remove (blr)

Removes the bundle lock that you specify by its number. You can find the number for each lock by using the `bundle-lock-list` command.

An example to remove all bundle locks:

```
zac blr --remove-all
```

bundle-props (bp)

Returns property information for a bundle.

An example to get the bundle properties of a specified bundle:

```
zac bp MacBundle
```

bundle-refresh (br)

Refreshes information about the specified bundle.

An example to refresh a bundle:

```
zac br MacBundle
```

bundle-search (bse)

Searches for bundles matching the specified java regular expression or string.

bundle-uninstall (bu)

Uninstalls the specified bundle. The actions that support the *Undo* action are executed while uninstalling the bundle. After the bundle uninstall is complete, the bundle state is displayed as `Not Installed`.

Use the `bundle-list` command to get a list of the installed bundles and their display names.

An example to uninstall a bundle:

```
zac bu MacBundle
```

bundle-verify (bv)

Reinstalls the bundle and executes the *Verify* action sets to ensure that no files have been removed or corrupted. Use the `bundle-list` command to get a list of the installed bundles and their display names.

An example to verify a bundle:

```
zac bv MacBundle
```

Core Commands

add-reg-key (ark)

This command registers a device by using the specified registration key.

Example: `zac add-reg-key`

agent-properties (agp)

This command provides information about the Adaptive Agent.

Example: `zac agent-properties`

cache-clear (cc)

Clears the cache on this device.

Example: `zac cc`

cert-info (ci)

Lists public key certificate information for each known ZCM server or adds a trusted root certificate to the machine store.

Example:

To list the certificate for each known ZENworks user:

```
zac ci
```

To add a trusted root certificate to the devices trusted store:

```
zac ci c:\certs\mytrustcacert.der -u myuser -p mypassword
```

config-location (cl)

Displays the current location.

Example:

```
zac config-location
```

file-system-guid (fsg)

Removes and restores the workstation GUID in the file system in preparation for taking an image.

Example:

```
zac fsg
```

get-pref (get)

Displays the preferences that are currently set.

Example to display all preferences that have names beginning with abc:

```
zac get-pref abc
```

get-settings (gs)

Returns the effective settings associated with the given key.

Example to list the Remote Management settings:

```
zac gs RemoteManagement
```

logger (log)

This command changes the logger configuration.

Example to reset the log file:

```
zac logger resetlog
```

reestablish-trust (retr)

This command re-establishes trust with the current zone.

Reestablishes trust with the current Management Zone. The username and password used must be of the Zone Administrator.

Example:

```
zac retr -u myuser -p mypassword
```

The -u and -p parameters are optional. If you don't use the -u and -p parameters, you are prompted to enter a username and password.

refresh (ref)

This command triggers a refresh of the specified type.

register (reg)

This command registers a device to a specified zone.

Example:

```
zac reg -k key1 https://123.456.78.90
```

reregister (rereg)

This command registers a device in the current zone and assigns it to the GUID of an existing device object. The currently associated device object is deleted.

To reregister, specify a username and password:

```
zac reregister -u myuser -p mypassword eaa6a76814d650439c648d597280d5d4
```

set-pref (set)

Sets the value for a preference.

Example to set the value of preference abc to 123:

```
zac set-pref abc 123
```

set-proxy (sp)

Sets the proxy that the ZENworks Agent will use.

Example:

```
zac sp 123.456.78.90 administrator novell
```

unregister (unr)

This command unregisters a device from the current zone.

Example:

```
zac unr -u administrator -p novell
```

zone-config (zc)

This command displays the Zone Configuration server or lists the current zone configuration.

```
zac zc
```

Inventory Commands

inventory (inv) [scannow]

Runs an inventory scan.

An example to run an inventory scan:

```
zac inv scannow
```

Local User Management Commands

User-add (ua)

Adds a local ZMD user.

An example to add a ZMD user:

```
zac ua <UserName>
```

user-delete (ud)

Deletes a local ZMD user.

An example to delete a ZMD user:

```
zac ud <UserName>
```

user-edit (ue)

Edits the local ZMD user privileges.

An example to edit the ZMD user privileges:

```
zac ue <UserName> [+/-Privilege]
```

user-list (ul)

Displays local ZMD users and their privileges.

user-privilege-export (upe)

Exports user privileges to the specified file.

Example:

```
zac upe <FileName>
```

user-privilege-import (upi)

Imports user privileges from the specified file.

Example:

```
zac upi <FileName>
```

Patch Management Commands

patch-scan (ps)

Scans the device for missing patches.

An example to run a patch scan:

```
zac ps
```

Satellite Roles Commands

satellite-cachecfg (satcfg)

Reads Satellite roles from cache.

An example to read Satellite roles from cache:

```
zac satcfg
```

satellite-clearcache (satcc)

Clears Satellite configuration information from cache.

An example to clear Satellite configuration information from cache:

```
zac satcc
```

satellite-refreshcfg (satrc)

Refreshes Satellite roles.

An example to refresh Satellite roles:

```
zac satrc
```

satellite-roles (satr)

Reports on current Satellite roles and their state.

An example to report on current Satellite roles and their state:

```
zac satr
```

Status Commands

- "zenhttp-status(zhs)" on page 148

zenhttp-status(zhs)

Lists port and tags for registered handlers.

An example to list port and tags for registered handlers:

```
zac zhs
```

This command is applicable only if the agent is promoted as a satellite.

Authors

Copyright 2008-2012, Novell, Inc. All rights reserved. http://www.novell.com

2 ZENworks Diagnostic Center

Novell ZENworks 11 provides a tool called as ZENworks Diagnostic Center that helps you to verify the health of the Primary Servers and the Management Zone before deploying a system update.

ZENworks Diagnostic Center (ZDC) performs a series of diagnostic tests on Primary Servers, and determines the state of the Primary Servers and the Management zone. ZDC comprises of the following verifiers:

- **ZENworks System Files Verifier:** Verifies the existence and consistency of critical system files that are required by ZENworks 11.
- **ZENworks Database Schema Verifier:** Validates the correctness of the schema (tables, indexes, database procedures, etc) of the database used by ZENworks 11.
- **ZENworks Content Verifier:** Validates the existence and consistency of the contents hosted on the Primary Server that has ZDC installed and running.

To verify the database schema before upgrading to ZENworks 11, you must use the ZENworks Diagnostic Center tool available in the ZENworks 11 Installation media. To verify the health of the Primary Servers and the Management Zone after upgrading to ZENworks 11, download and use the ZENworks Diagnostic Center from ZENworks 11 ZENworks Control Center. For more information, review the following sections:

2.1 Downloading and Extracting ZDC

You need to download and extract ZENworks Diagnostic Center on the Primary Server on which you want to run the tool.

1. Open a Web browser to the Novell Download (http://download.novell.com/patch/finder/#familyId=18962&keywords=ZDC) site.
2. Select the specific version of ZENworks Configuration Management in the *Standard Select* drop-down list.
3. Select the appropriate ZENworks Diagnostic Center patch listed in the ZENworks Configuration Management panel.
4. From the ZCM <version> ZENworks Diagnostic Center - executable and pattern file page, download the `ZENworksDiagnosticCenter.zip` file to a temporary location and extract the ZIP file contents.

   ZENworks system files and the ZENworks database schema require baseline data collected from known good deployments to compare with and assess the health of the current deployment data.

The ZIP file contains the baseline data file, `baseline_<version>.zdc`, which is available in the `data` directory. This file contains the baselines data collected from the known good deployments of ZENworks 11 Configuration Management SP2 (11.2.2).

ZDC executable and configuration files are required to run ZDC. The directory that contains the ZENworks Diagnostic Center is referred to as `$ZDC_Home`.

2.2 Running ZDC

1 At Primary Server's console prompt, go to the `$ZDC_Home` directory.

2 Run the appropriate command.

- ◆ To verify the ZENworks system files, the ZENworks database schema, and the ZENworks content:

   On Windows: `zdc.bat verify -v systemfiles,zendatabase -b <complete_path_of_the_ZDC_baseline_data_file>`

   On Linux: `./zdc verify -v systemfiles,zendatabase -b <complete_path_of_the_ZDC_baseline_data_file>`

- ◆ To verify only for the ZENworks system files:

   On Windows: `zdc.bat verify -v systemfiles -b <complete_path_of_the_ZDC_baseline_data_file>`

   On Linux: `./zdc verify -v systemfiles -b <complete_path_of_the_ZDC_baseline_data_file>`

- ◆ To verify only for the ZENworks database schema:

   On Windows: `zdc.bat verify -v zendatabase -b <complete_path_of_the_ZDC_baseline_data_file>`

   On Linux: `./zdc verify -v zendatabase -b <complete_path_of_the_ZDC_baseline_data_file>`

- ◆ To verify only for the ZENworks content:

   On Windows: `zdc.bat verify -v zencontent`

   On Linux: `./zdc verify -v zencontent`

You can also specify the following command options while running ZDC.

-s: session name
-l: complete path of the ZDC log filename
-r: complete path of the ZDC `report` directory

ZDC runs the diagnostic tests and generates the reports in the HTML format. The reports are stored in the `$ZDC_Home/reports/`*timestamp* directory on a Windows Primary Server and in the `$ZDC_Home\reports\`*timestamp* directory on a Linux Primary Server. To view the reports, open `index.html` that is located in the `report` directory.

The complete log of the diagnostic tests is stored in `$ZDC_HOME/logs/`*timestamp.log* on a Windows Primary Server and in the `$ZDC_HOME\logs\`*timestamp.log* on a Linux Primary Server.

3 Troubleshooting ZENworks Command Line Utilities

The following sections provide solutions to the problems you might encounter while working with Novell ZENworks 11 command line utilities:

- "zman hangs when the output of a command is redirected to a file by using the command line redirection operators such as > or >>" on page 151
- "An incorrect success message is displayed when the "stage group name" optional argument is not passed to the system-update-staging-group-create command" on page 152
- "Unable to save the report on SLES 10 SP1 by using the zman rpgn command" on page 152
- "The Imaging role added to a Satellite by using the zman ssaimg command is not immediately applied to the server" on page 152
- "The zman commands do not work for non-root users on a ZENworks Linux server" on page 152
- "A non-root user is unable to configure a Satellite role by using the zman ssc command" on page 153
- "Unable to use the zman sui command to import the ZENworks system update to a ZENworks Server, on an RHEL-5.x Server" on page 153

zman hangs when the output of a command is redirected to a file by using the command line redirection operators such as > or >>

Source: ZENworks 11 ; zman.

Possible Cause: zman requires the ZENworks administrator username and password for executing each command. If the credentials are not provided with the command, you are prompted to enter the same. However, if the output of the command is redirected to file, you are not prompted to enter the credentials because the prompts for username and password are also redirected to the file. Consequently, zman hangs, waiting for the credentials to be entered.

Action: Specify the credentials as part of the command by using `-U|--User` and `-P|--Password` options.

Action: Store the credentials by using the `admin-store-credential` command and then execute the command.

Action: Redirect the output to a file by using the `-R|--Redirect` option instead of a redirection operator such as > or >>.

An incorrect success message is displayed when the "stage group name" optional argument is not passed to the system-update-staging-group-create command

Source: ZENworks 11 ; zman.

Explanation: zman displays the following incorrect message if the stage group name optional argument is not passed to the `system-update-staging-group-create` command:

`Successfully created the object "null"`

Action: To view the list of staging groups that would include the newly created staging group, execute the `system-update-staging-group-list (susgl)` command at the console prompt.

Unable to save the report on SLES 10 SP1 by using the zman rpgn command

Source: ZENworks 11 ; zman.

Action: None.

The Imaging role added to a Satellite by using the zman ssaimg command is not immediately applied to the server

Source: ZENworks 11 ; zman.

Explanation: If you configure the Imaging role by the using the `zman ssaimg` command, the role is immediately added to the device. If the role is not immediately added, it is added only during the next device refresh schedule.

Action: If you want to immediately apply the role to the device, you must manually refresh the device in one of the following ways:

- At the command prompt, run the `zman workstation-refresh|wrf` command. If the managed device is a server, run the `zman server-refresh|srf` command.

- On the managed device, right-click the icon, then click *Refresh*.

The zman commands do not work for non-root users on a ZENworks Linux server

Source: ZENworks 11 ; zman.

Explanation: If a non-`root` user executes the zman commands on a ZENworks Linux server, the following error message is displayed:

`You do not have sufficient privileges to run zman commands. As a non-root user in Linux, you should be part of the zmanusers group to run zman commands.`

Action: Add the non-root user to the zmanusers group before executing the zman commands.

    1 Log in to the ZENworks server as root.

    2 At the server console prompt:

        ◆ If the user already exists on the Linux device, execute the following command:

        `groupmod -A non-rootusername zmanusers`

        ◆ If the user does not exist on the Linux device, execute the following command:

        `useradd -G zmanusers non-rootusername`

    3 Log in to the ZENworks server as a non-root user.

A non-root user is unable to configure a Satellite role by using the zman ssc command

Source: ZENworks 11 ; zman.

Explanation: If a non-root user tries to configure a Satellite role by using the zman ssc command on a Linux Primary Server, the following error message is displayed:

```
The role you specified (Satellite_role_name) is not a valid
satellite server role. Valid roles are:

ZoneConfig.ROLE_COLLECTION
ZoneConfig.ROLE_CONTENT
ZoneConfig.ROLE_AUTHENTICATION
```

Action: Do the following:

    1 Ensure that the non-root has been added to the zmanusers group.

      1a Log in to the ZENworks server as root.

      1b At the server console prompt:

        ◆ If the user already exists on the Linux device, execute the following command:

        `groupmod -A non-rootusername zmanusers`

        ◆ If the user does not exist on the Linux device, execute the following command:

        `useradd -G zmanusers non-rootusername`

    2 Add the non-root user to the ZENworks user group by using the `usermod -A non-root_username` command. Subsequently, the non-root user can configure Satellite roles.

Unable to use the zman sui command to import the ZENworks system update to a ZENworks Server, on an RHEL-5.x Server

Source: ZENworks 11; zman.

Explanation: On an RHEL-5.x server, if the size of the system update zip file is more than 2 GB, unzipping does not work, and you cannot import the system update by using the zman sui command.

Action: Perform the following steps:

1 Navigate to the Red Hat vvitek directory (http://people.redhat.com/vvitek/unzip-5.52-3.0.bz497482.el5_7/).

2 Install `unzip-5.52-3.0.bz497482.el5_7.i386.rpm` on the RHEL 5.x server.

3 Run the `rpm -Uvh unzip-5.52-3.0.bz497482.el5_7.i386.rpm` command.

4 Run the `zman sui <zip file path>` command.

4 Guidelines for Working with Zman

Follow these guidelines as you work with the zman command line utility of Novell ZENworks 11 :

- If an XML file that is exported by using the zman commands such as `bundle-create` or `policy-create` contains diacritic or extended ASCII characters such as ñ,ë, and Ä, you must open the file in an editor using the UTF-8 encoding. If you make any changes to the file, you must save it the UTF-8 format.

- If you want to redirect the output of a command containing extended ASCII characters to a file on Windows, you must not use the command line redirection operator (>), because the code page of an MS-DOS prompt is different from the code page used to write to files on most locales.

  For example, in case of Western European languages like English, French, German, and Spanish, the code page of the DOS prompt is cp437 or cp850, but the rest of the Windows operating system uses cp1252.

  To redirect the output to a file and correctly retain the extended ASCII characters, use the `-R|--Redirect` global option.

- While connecting to Linux Servers from a Windows machine by using clients like PuTTY, set the character set used for translation to UTF-8. This ensures proper translation of characters other than standard ASCII characters.

  To set the character set for translation to UTF-8 in PuTTY:

  1. Open the PuTTY client.

  2. In the PuTTY Configuration window, click *Windows > Translation*.

  3. In the *Received Data Assumed to be in Which Character Set* drop-down list, select *UTF-8*.

  NOTE: You do not need to do this if the input to the command or its output contains only the a-z or A-Z characters.

- By default, zman uses the default locale of the server. You can have zman use a specific language:

  - On Windows, specify the appropriate user language and file encoding as a value for the JVM_STARTUP_OPTIONS property in `ZENworks_Installation_directory\Novell\ZENworks\conf\zman\properties\zman-config.properties`.

    For example, to run zman in English, set the value of `JVM_STARTUP_OPTIONS` to `-Duser.language=en -Dfile.encoding=cp850`. After editing the file, change the code page of the command prompt to `cp850` by using the `chcp 850` command.

  - On Linux, edit `/opt/novell/zenworks/bin/zman` to add `-Duser.language=<language>` after `/opt/novell/zenworks/lib/java/bin/java`.

    For example, to run zman in English, change the zman script file as follows:

    ```
    /opt/novell/zenworks/lib/java/bin/java -Duser.language=en -
    Djava.library.path="${LD_LIB_PATH}" .....
    ```

A Documentation Updates

This section contains information on documentation content changes that were made in this *Command Line Utilities Reference* for Novell ZENworks 11. The information can help you to keep current on updates to the documentation.

The documentation for this product is provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the changes listed in this section.

If you need to know whether a copy of the PDF documentation that you are using is the most recent, the PDF document includes a publication date on the title page.

The documentation was updated on the following dates:

A.1 March 2013: Update for ZENworks 11 SP2 (11.2.3)

Location	Change
Chapter 1, "ZENworks Command Line Utilities," on page 7	Included a new option under the following: "bundle-create (bc) (bundle name) (bundle XML file) [parent folder] [options]" on page 19.