Novell
ZENworks® 10 Configuration Management

10.0.3

May 16, 2008

COMMAND LINE UTILITIES
REFERENCE

Novell®

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

This *Novell ZENworks 10 Configuration Management Command Line Utilities Reference* includes information to help you use the following Novell® ZENworks® utilities:

- zman(1) (page 10)
- zeninfocollect(1) (page 60)
- zac(1) (page 62)

The information in this guide is organized as follows:

- Chapter 1, "ZENworks Command Line Utilities," on page 9
- Chapter 2, "Troubleshooting ZENworks Command Line Utilities," on page 67
- Chapter 3, "Guidelines for Working with Zman," on page 69

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, or go to the Novell Documentation Feedback site (http://www.novell.com/documentation/feedback.html) and enter your comments there.

Additional Documentation

ZENworks 10 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 10 Configuration Management documentation (http://www.novell.com/documentation/zcm10/index.html).

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux*, should use forward slashes as required by your software.

ZENworks Command Line Utilities 1

The following sections for information on the command line utilities:

-
-
-

zman(1)

Name

zman - zman is the command-line interface to Novell® ZENworks® 10 Configuration Management, which provides comprehensive server, workstation, and user management.

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 a large number of commands, but you will find that most of them are easy to remember, since 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` whereas `bcp` is the short form of `bundle-copy`.

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

Options

Options are of two types - switch and flagged option. 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 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, in bundle-list specifying the –recursive switch tells that the contents of the sub folders 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 mostly ordered in the same way as the command itself: for `server-add-bundle`, the server is named first and the bundle second.

Description

Novell ZENworks 10 Configuration Management is the next generation in server, workstation, and user management. It provides comprehensive resource management including:

- Advanced software management.

- Discovery of deployable devices on the network and deployment of ZENworks software to them
- Secure and fast graphical remote management of servers and desktops.
- Hardware, software, and Operating System inventory collection and reporting.
- Comprehensive policy-based management of servers, desktops, and users.
- Task-driven web management interface.

The zman command-line interface provides you a full-featured application that can simplify many operations using scripting and provide quick access to operations. It can be used effectively in automation of tasks. ZENworks Control Center, a comprehensive Web management interface with many advance features is also installed on your ZENworks server.

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 four levels. Pipeline the help through the operating system's native 'more' command for scrollable output.

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

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

  Example: `zman -h`
- At 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 which 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 normal braces ( ) are mandatory and those enclosed by square braces [ ] are optional.

Ellipsis (...)

Ellipsis indicates that a command accepts multiple entries of the last argument type. The arguments will have to be separated by space or commas depending on the command. For example, the ellipsis in following command:

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

indicate 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 a `-f|--folder` option. This option provides convenience 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 is to be specified relative to the root folder of that object separated by a forward slash '/', unless 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.

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

Stores the ZENworks username and password for use with every command. This way credentials need not be entered for every command. The credentials are stored on the home directory of the logged in user and will be 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, it is must that you clear your credentials using admin-clear-credential command after you are done executing commands.

admin-clear-credential (acc)

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

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

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 will prompt for the password of the administrator, to avoid entering the password in clear text. If the administrator account is based on a user, it will use the same credentials defined in the user source and hence the command will not prompt for password.

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

Renames a ZENworks administrator account.

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

Deletes ZENworks administrator accounts.

(ZENworks administrator name) [...] - Name of the ZENworks administrators. 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-rights-set (ars) [options] [ZENworks administrator name] [rights category] [object path] [...]

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 objects object type on 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 of 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, path of the objects are 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 long and short form of right names within each category. If a category name is specified, help is provided for that category.

admin-rights-delete (ard) [options] (ZENworks administrator name) [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 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`.

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

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`.

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

Batch Commands

This is a utility command used to execute a list of zman commands from a text file. The commands are executed faster as 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.

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.

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

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

(file path) [...] - 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_directory:\Novell\Zenworks\share\zman\samples\batch` on a Windows server.

Accepts the following options:

-e, --exit-on-failure - Stops executing the commands and exits if a command fails partially or totally. A command is said to partially fail in cases of commands that take multiple arguments when the processing of one or more of the arguments was not successful. For example, in case of 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.

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

Copies a bundle. Accepts the following options:

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

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

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

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 a XML file. Template XML files can be created from bundles created through ZENworks Control Center and then reused. 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 dependency. These data need to be specified in addition 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] - Path of the image file containing the image that should be used as the bundle icon.

bundle-export-to-file (betf) (bundle path) (XML file path)

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

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

Adds actions to a bundle.

(action XML file) [...] - The XML files that contain information of actions to be added. The actions are grouped under one of these action sets: Install, Launch, Verify, Uninstall, Terminate, Preboot. The XML file will contain 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 bundle-export-actions command.

Accepts the following options:

-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. These data need to be specified in addition 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.

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

Removes actions from a bundle.

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

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

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) - Type of the action set which contains the action. Valid values are "Install", "Launch", "Verify", "Uninstall", "Terminate", and "Preboot".

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

Lists actions added to a bundle.

[action set type] - 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".

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) - 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".

[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 and if not specified, all the actions in the specified action set are exported.

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

Creates a new folder for containing bundles. Accepts the following options:

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

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

Creates a bundle group and adds members to it. Accepts the following options:

--desc=[description] - Description for the group.
-m, --members=[bundle path][...] - Path of the bundles relative to /Bundles.
-f, --folder=[bundle folder] - 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-add (bga) [options] (bundle group path) (bundle path) [...]

Adds bundles to a bundle group. Accepts the following options:

-f, --folder=[bundle folder] - 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-remove (bgr) [options] (bundle group path) (bundle path) [...]

Removes bundles from a bundle group. Accepts the following options:

-f, --folder=[bundle folder] - 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) [options] (bundle group path)

Lists members of a bundle group. Accepts the following options:

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

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

Moves a bundle object to a different folder.

bundle-rename (br) (bundle object path) (new name)

Renames a bundle object.

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

Deletes one or more bundle objects.

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

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

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

(device or user object path) [...] - 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] - 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.
-L, --launch-immediately - Launches the bundle immediately after installation.

-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 using Wake-On-LAN if it is shut down while distributing the bundle.

For more information on the schedules and how to specify the schedule in an XML file format, refer to the Schedules section of the ZENworks Administration Guide.

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

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) [...] - 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] - 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-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) [...] - Path of the device or user objects relative to which the bundle group is assigned.

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

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) [...] - 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] - 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 has to be removed.

-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 has to be removed.

-l, --launch-schedule=[launch schedule XML file or "NoSchedule"] - XML file that contains
the launch schedule or "NoSchedule" if the schedule has to removed.

-a, --availability-schedule=[availability schedule XML file or "NoSchedule"] - XML file that
contains the availability schedule or "NoSchedule" if the schedule has to 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] - Set the distribution schedule to distribute the
bundle on device refresh, or remove 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] - Set 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] - Wake up the device using Wake-On-LAN if
it is shut down while distributing the bundle, or remove the same option set during bundle
assignment. Valid values are "true" or "yes", and "false" or "no".

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

Lists the device and user assignments for a bundle. Accepts the following options:

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

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

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

Enables bundles. Accepts the following options:

-f, --folder=[bundle folder] - 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-disable (bdl) [options] (bundle path) [...]

Disables bundles. Accepts the following options:

-f, --folder=[bundle folder] - Path of a bundle folder relative to /Bundles. If this option is specified, 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) [options] (bundle path) [...]

Increments the version of bundles. Accepts the following options:

-f, --folder=[bundle folder] - 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-view-summary-totals (bvst) (bundle path)

Displays summary of the deployment status of a bundle.

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

Displays advanced deployment status of a bundle. Accepts the following options:

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

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

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

--statusfilter=[status type][...] - Filter 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][...] - Filter 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.

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) [options] (file path)`

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

-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-enable (care)`

Enables the Certificate Authority role on the local server.

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

Disables the Certificate Authority role on the local server.

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.

`database-get-credentials (dgc)`

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

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

Backs up the embedded Sybase SQL Anywhere database and also allows scheduling of the backup operation.

(backup directory) - The local directory on the database server 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 options:

-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 is run on Tuesday with the backup directory specified as "c:\" and the value for this option specified as DAYNAME(today()), then the files are backed up to c:\Tuesday.

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

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

(user credential file) - 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 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 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.

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.

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

Deletes the deployment task.

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

Aborts the deployment task.

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]`

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 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 will be 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 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.

Hotlist Commands

This command is used to view the list of devices which have unacknowledged warnings or errors. Hotlist commands 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 which have unacknowledged warnings or errors. Accepts the following options:

-c, --count - Display only the count of the hotlist items.

-t, --type=[error type] - Filter 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 comparison operator is not specified ">=" 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 using the messages-acknowledge command. Once all warnings or errors for a device are acknowledged it no longer appears on the hotlist.

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

Initiates an inventory scan of one or more devices. Accepts the following options:

-f, --folder=[device folder] - 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 can be used to specify multiple device objects under the same folder.

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

Sends the inventory data collection form to one or more devices. Accepts the following options:

-f, --folder=[device folder] - 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 can be used to specify multiple device objects under the same folder.

License Commands

These commands are used to activate your ZENworks server or 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-info (li)`

Displays the licensing information.

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

Activates the ZENworks products.

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

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

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

-t, --type=[message type] - Filter 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] - Filter messages based on 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 specified, 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 - Filter on acknowledged messages.

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

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

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

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

-A, --asc - Specify the direction of sorted listing. Default is descending order.

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

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

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

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

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

Accepts the following options:

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

-e, --end-date=[message logged date] - Messages logged on and before this date will be 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 using the messages-view command.

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

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 a policy.

Accepts the following options:

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

-e, --end-date=[message logged date] - Messages logged on and before this date will be 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 using the `messages-view` command.

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

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)

Gets the GUID of a ZENworks object.

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

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-list (pl) [options] [folder]

Lists policy objects. Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, then 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-copy (pcp) [options] (policy path) (policy name) [parent folder]

Copies a policy. Accepts the following options:

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

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

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. Template XML files can be created from policies created through ZENworks Control Center and then reused.

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.

policy-export-to-file (petf) (policy path) (XML file path)

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

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

Creates a new folder for containing policies. Accepts the following options:

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

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

Creates a policy group and adds members to it. Accepts the following options:

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

-m, --members=[policy path][...] - Path of the policies relative to /Policies.

-f, --folder=[policy folder] - 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-add (pga) [options] (policy group path) (policy path) [...]

Adds policies to a policy group. Accepts the following options:

-f, --folder=[policy folder] - 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-remove (pgr) [options] (policy group path) (policy path) [...]

Removes policies from a policy group. Accepts the following options:

-f, --folder=[policy folder] - 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) [options] (policy group path)

Lists members of a policy group. Accepts the following options:

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

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

Lists groups of which the given policy is a member. Accepts the following options:

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

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

Moves a policy object to a different folder.

policy-rename (pr) (policy object path) (new name)

Renames a policy object.

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

Deletes one or more policy objects.

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

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

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

Lists the device and user assignments for a policy. Accepts the following options:

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

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

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

Enables policies. Accepts the following options:

-f, --folder=[policy folder] - 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) [options] (policy path) [...]

Disables policies. Accepts the following options:

-f, --folder=[policy folder] - 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-increment-version (piv) [options] (policy path) [...]

Increments the version of policies. Accepts the following options:

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

policy-view-summary-totals (pvst) (policy path)

Displays summary of the deployment status of a policy.

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

Displays advanced deployment status of a policy. Accepts the following options:

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

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

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

--statusfilter=[status type][...] - Filter 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][...] - Filter 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.

Queue Commands

These commands are used in situations 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-list (ql) [options] [server path]

Lists queue entries.

[server path] - 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] - Filter on the type of the queue entry. For example, "content.cleanup" is a queue entry type.

-s, --status=[queue status] - Filter on the status of the queue entries. Valid values are "N", "I", "C", "S", and "F". They stand for New, In Progress,Cancelled, Success, and Failed respectively.

queue-reset (qr) [server path]

Resets the queue, setting the status of failed queue entries to new.

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

queue-flush (qf) (queue status)

Flushes the queue by removing the queue entries.

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

Registration Commands

These commands allow you to create and alter registrations. The commands require a device type to be specified. It determines the type of device folder and device group the registration key will be associated with and hence 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-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, then 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-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, then 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-create-key (rck) [options] (device type) (registration key) [registration folder path]

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 will be 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 will become a member of these device groups.

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

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

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

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

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

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

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 will be placed in this folder.

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

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

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

--loc=[location] - The physical location of the devices. Registering devices will be 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-folder-create (rfc) [options] (folder name) [parent folder]`

Creates a new registration folder. Accepts the following options:

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

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

Moves a registration object to a different registration folder.

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

Renames a registration object.

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

Deletes registration objects.

(registration object path) [...] - 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] - Path of a registration folder relative to /Keys. If this option is specified, path of the registration objects are determined relative to this folder. This can be used to specify multiple registration objects under the same folder.

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

Lists the device groups associated with a registration key.

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

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 options:

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

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

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 options:

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

Report Commands

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

report-list (rpl) [options] (report type)

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, then 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, then the content of the root folder is listed.

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

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-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-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-deploy (rpd) (file path) (destination folder) [report type]

Deploys reports to the BusinessObjects Enterprise Server.

The reports are created on the Business Object Enterprise Server with the same name as the report file.

(destination folder) - The folder to which reports are to be deployed.

[report type] - Filter 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, then only the Web Intelligence* documents in the folder are deployed.

report-generate-now (rpgn) [options] (report name) (report folder) (report type) (format type) (report destination)

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] - 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) [options] (report name) (report folder) (report type) (destination folder) (export format)

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. 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-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-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) [options] (report name) (report folder) (report type)

>Deletes the report instances from 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-add-user (rpau) (Username)

>Adds a new user account to the Enterprise Server.
>
>(Username) - The name of the user to be added to the BusinessObjects Enterprise Server.

report-folders-sync (rpfs)

>Synchronizes the report folders in the BusinessObjects Enterprise Server with ZENworks.

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. It determines the type of device folder and device group the rule set will be associated with and hence 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-list (rsl)`

> Lists all rule sets.

`ruleset-info (rsi) (name or position)`

> Displays detailed information about a rule set.

`ruleset-create (rsc) [options] (device type) (rule set name) (rules file) [position]`

> 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."
>
> (rules file) - Path of the file containing the rules to be added. For more information on the format of the contents of the file, refer to the `ruleset-add-rule` command.
>
> 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 will be 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 will become a member of these device groups.
>
> --desc=[description] - Description for the rule set.
>
> --site=[site] - The site where the devices are located. Registering devices will be populated with this site information.
>
> -dept=[department] - The department in which the devices are used. Registering devices will be populated with this department information.
>
> --loc=[location] - The physical location of the devices. Registering devices will be populated with this location information.

`ruleset-update (rsu) [options] (device type) (name or position)`

> Updates a rule set. 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 will be placed in this folder.
>
> --desc=[description] - Description for the rule set.
>
> --site=[site] - The site where the devices are located. Registering devices will be populated with this site information.
>
> -dept=[department] - The department in which the devices are used. Registering devices will be populated with this department information.
>
> --loc=[location] - The physical location of the devices. Registering devices will be populated with this location information.

ruleset-move (rsmv) (name or position) (new position)

Changes the position of a rule set.

ruleset-rename (rsr) (name or position) (new name)

Renames a rule set.

ruleset-delete (rsd) (name or position)

Deletes a rule set.

ruleset-add-rule (rsar) (name or position) (rules file)

Adds a rule to a rule set.

(rules file) - The path of the file containing the rules to be added. Each rule defines a condition which must be met in order for a rule set to be applied to a device. Each rule in a rule set must evaluate to TRUE in order for the rule set to be applied. A rule is of the following format:

NOT attribute operator value

NOT is used to indicate rule negation and is optional. It must always be specified before the attribute.

attribute - The valid attributes are "GUID", "DeviceType", "OS", "HostName", "IPAddress", "CPU", "DNS", and "Language".

operator - Valid operators are Contains, StartsWith, EndsWith and Equals.

value - The format and contents of value are determined by the attribute. For a sample, 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.

ruleset-remove-rule (rsrr) (name or position) (rule position)

Removes a rule from a rule set.

ruleset-add-device-groups (rsadg) [options] (device type) (name or position) (device group path) [...]

Adds membership in the specified device groups for devices imported using the specified rule set.

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

Accepts the following options:

-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-remove-device-groups (rsrdg) [options] (device type) (name or position) (device group path) [...]

Removes the association of device groups to a rule set.

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

Accepts the following options:

-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-list-device-groups (rsldg) (name or position)

Lists the device groups associated with a rule set.

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-list (sl) [options] [folder]

Lists server objects. Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, then 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.

server-info (si) (server path)

Lists detailed information about a server.

server-statistics (sst) (server path)

Displays statistics of ZENworks server usage.

server-folder-create (sfc) [options] [folder name] [parent folder]

Creates a new folder for containing servers. Accepts the following options:

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

server-group-create (sgc) [options] [group name] [parent folder]

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-add (sga) [options] (server group path) [server path] [...]

Adds servers to a server group. Accepts the following options:

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

server-group-remove (sgr) [options] (server group path) [server path] [...]

Removes servers from a server group. 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.

server-group-members (sgm) [options] (server group path)

Lists members of a server group or a dynamic server group. Accepts the following options:

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

server-list-groups (slg) [options] (server path)

Lists groups of which the given server is a member. Accepts the following options:

-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-rename (sr) (server object path) (new name)

Renames a server object.

server-delete (sd) [options] (server object path) [...]

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-refresh (srf) (server object path) [...]

Refreshes the ZENworks agent in the servers.

server-add-bundle (sab) [options] (server object path) (bundle or bundle group) [...]

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.

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

-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 using Wake-On-LAN if it is shut down while distributing the bundle.

For more information on the schedules and how to specify the schedule in an XML file format, refer to the Schedules section of the ZENworks Administration Guide.

server-remove-bundle (srb) [options] (server object path) (bundle or bundle group) [...]

Removes bundles assigned 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.

server-reorder-bundles (srob) (server object path) (current position) (new position)

Changes the order of bundles assigned to a server. Use the command server-list-bundles to get the order of assigned bundles.

server-list-bundles (slb) [options] (server object path)

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-add-policy (sap) [options] (server object path) (policy or policy group) [...]

Assigns policies to a server object. 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=[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-remove-policy (srp) [options] (server object path) (policy or policy group) [...]

Removes policies assigned to a server object. Accepts the following options:

-f, --folder=[policy folder] - 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-reorder-policies (srop) (server object path) (current position) (new position)

Changes the order of policies assigned to a server. Use the command `server-list-policies` to get the order of assigned policies.

server-list-policies (slp) [options] (server object path)

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-view-all-policies-status (svaps) [options] (server path)

Displays the advanced deployment status of policies assigned to a server. Accepts the following options:

--statusfilter=[status type][...] - Filter 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-refresh (srf) [options] (server object path) [...]

Refreshes the ZENworks Adaptive Agent in the servers. Accepts the following options:

-f, --folder=[server folder] - 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-wakeup (sw) [options] (server object path) [...]

Wakes up a server using Wake-On-LAN. Accepts the following options:

-f, --folder=[server folder] - 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.

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.

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-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 will be exported.

Accepts the following options:

-s, --source=[source device or device folder path] - Path of the device or device folder relative to /Devices from which settings have to be exported. If this option is not specified, settings will be exported from the Zone.

-e, --effective - If specified, the effective settings will be retrieved, else only the settings defined/overridden at the source path will be retrieved.

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 will be exported.

Accepts the following options:

-s, --source=[source device or device folder path] - Path of the device or device folder relative to /Devices for which settings have to be set. If this option is not specified, settings will be set at the Zone level.

settings-copy (scp) [options] (source device or device folder path) (destination device or device folder path) [...]

Copies settings.

(source device or device folder path) - Path of the device or device folder relative to /Devices from which settings have to be copied.

(destination device or device folder path) [...] - Path of the devices or device folders relative to /Devices to which settings have to be copied.

Accepts the following options:

-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 will be copied.

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) - Path of the device or device folder relative to /Devices, whose settings has to be reverted.

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-ex-view-available (sueva) [options]

Displays a list of all available updates or PRUs published 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 for which to check for updates (for example, zcm, pru).
-v, --version=[product base version] - The base version of the product for which to check for updates (for example, 10.0, 10.1).

system-update-ex-download (sued) (system update or PRU name)

Downloads the latest system updates or PRU.

(system update or PRU name) - System update/PRU name or UID.

system-update-ex-status (sues) (system update or PRU name)

Displays the status of the specified system update or PRU.

(system update or PRU name) - System update/PRU name or UID.

system-update-ex-cancel-download (suecd) (system update or PRU name)

Cancels the download of the system update or PRU.

(system update or PRU name) - System update/PRU name or UID.

system-update-ex-assign (suea) (system update name) [device path] [...] [options]

Assigns the authorized system update to devices.

(system update name) - System update name or UID.

[device path] [...] - Path of the device relative to /Devices.

Accepts the following options:

-f, --folder=[device folder] - 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 provides convenience if you want to specify multiple device objects under the same folder.
-a, --all - Assign to all devices.

system-update-ex-list-assign (suela) (device path) [option]

Lists all updates assigned to the specified device or group.

Accepts the following option:

-f, --folder=[device folder] - Path of the device folder relative to /Devices. If this option is specified, the path of the device object is determined relative to this folder. This provides convenience if you want to specify multiple device objects under the same folder.

system-update-ex-authorize (sueaz) (system update or PRU name)

Authorizes an update or PRU.

(system update or PRU name) - System update/PRU name or UID.

system-update-ex-delete-update (sueda) (system update or PRU name) [option]

Deletes a specified system update or PRU from the zone.

(system update or PRU name) - System update/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-ex-baseline (sueb) (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-ex-import (suei) (path to update) [system update or PRU name]

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, they 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/PRU name or UID.

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-list (ul) [options] [folder]

Lists user objects. Accepts the following options:

-r, --recursive - Lists results recursively including subfolders. If this option is not specified, then 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-info (ui) (user path)

Lists detailed information about a user.

user-source-create (usc) [options] (user source XML file path)

Configures a user source. Accepts the following options:

 -a, --accept-certificate - Accept the certificate presented by the user source when SSL mode is chosen. This option is provided to avoid interactive mode in case of scripting. It is advisable to view the certificate and then accept it.

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) - Display name for the user container.

user-group-create (ugc) [options] (group name)

Creates a ZENworks user group and adds members to it. Accepts the following options:

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

-m, --members=[user path][...] - Path of the users relative to /Users.

-f, --folder=[user folder] - 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-add (uga) [options] (ZENworks user group path) (user path) [...]

Adds users to a ZENworks user group. Accepts the following options:

-f, --folder=[user folder] - 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-remove (ugr) [options] (ZENworks user group path) (user path) [...]

Removes users from a ZENworks user group. Accepts the following options:

-f, --folder=[user folder] - 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-members (ugm) [options] (user group path)

Lists members of a ZENworks user group or LDAP Directory user group. Accepts the following options:

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

user-list-groups (ulg) [options] (user path)

Lists groups of which the given user is a member. Accepts the following options:

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

user-add-bundle (uab) [options] (user object path) (bundle or bundle group) [...]

Assigns bundles to a user object. Accepts the following options:

-f, --folder=[bundle folder] - 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.

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

-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.

For more information on the schedules and how to specify the schedule in an XML file format, refer to the Schedules section of the ZENworks Administration Guide.

user-remove-bundle (urb) [options] (user object path) (bundle or bundle group) [...]

Removes bundles assigned to a user object. Accepts the following options:

-f, --folder=[bundle folder] - 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-reorder-bundles (urob) (user object path) (current position) (new position)

Changes the order of bundles assigned to a user.

user-list-bundles (ulb) [options] (user object path)

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-view-all-bundles-status (uvabs) [options] (user path)

Displays the advanced deployment status of bundles assigned to a user. Accepts the following options:

--statusfilter=[status type][...] - Filter 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-add-policy (uap) [options] (user object path) (policy or policy group) [...]

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] - 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-remove-policy (urp) [options] (user object path) (policy or policy group) [...]

Removes policies assigned to a user object. Accepts the following options:

-f, --folder=[policy folder] - 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-policies (urop) (user object path) (current position) (new position)

　　Changes the order of policies assigned to a user.

user-list-policies (ulp) [options] (user object path)

　　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-view-all-policies-status (uvaps) [options] (user path)

　　Displays the advanced deployment status of policies assigned to a user. Accepts the following options:

　　--statusfilter=[status type][...] - Filter 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.

user-logged-in-devices (ulid) (user path)

　　Displays the devices on which a user is logged in.

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-list (wl) [options] [folder]

　　Lists workstation objects. Accepts the following options:

　　-r, --recursive - Lists results recursively including subfolders. If this option is not specified, then 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.

workstation-info (wi) (workstation path)

　　Lists detailed information about a workstation.

workstation-folder-create (wfc) [options] (folder name) [parent folder]

Creates a new folder for containing workstations. Accepts the following options:

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

workstation-group-create (wgc) [options] (group name) [parent folder]

Creates a workstation group and adds members to it. Accepts the following options:

--desc=[description] - Description for the group.
-m, --members=[workstation path][...] - Path of the workstations relative to /Devices/ Workstations.
-f, --folder=[workstation folder] - 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-add (wga) [options] (workstation group path) (workstation path) [...]

Adds workstations to a workstation group. Accepts the following options:

-f, --folder=[workstation folder] - 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-remove (wgr) [options] (workstation group path) (workstation path) [...]

Removes workstations from a workstation group. Accepts the following options:

-f, --folder=[workstation folder] - 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 options:

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

workstation-list-groups (wlg) [options] (workstation path)

Lists groups of which the given workstation is a member. Accepts the following options:

-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-rename (wr) (workstation object path) (new name)

Renames a workstation object.

workstation-delete (wd) [options] (workstation object path) [...]

Deletes one or more workstation objects.

(workstation object path) [...] - 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] - 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-refresh (wrf) (workstation object path) [...]

Refreshes the ZENworks agent in the workstations.

workstation-add-bundle (wab) [options] (workstation object path) (bundle or bundle group) [...]

Assigns bundles to a workstation object. Accepts the following options:

-f, --folder=[bundle folder] - 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.

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

-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 using Wake-On-LAN if it is shut down while distributing the bundle.

For more information on the schedules and how to specify the schedule in an XML file format, refer to the Schedules section of the ZENworks Administration Guide.

workstation-remove-bundle (wrb) [options] (workstation object path) (bundle or bundle group) [...]

Removes bundles assigned to a workstation object. Accepts the following options:

-f, --folder=[bundle folder] - 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-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-list-bundles (wlb) [options] (workstation object path)

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-add-policy (wap) [options] (workstation object path) (policy or policy group) [...]

Assigns policies to a workstation object. 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=[policy folder] - 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-remove-policy (wrp) [options] (workstation object path) (policy or policy group) [...]

Removes policies assigned to a workstation object. Accepts the following options:

-f, --folder=[policy folder] - 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-reorder-policies (wrop) (workstation object path) (current position) (new position)`

Changes the order of policies assigned to a workstation. Use the command workstation-list-policies to get the order of assigned policies.

`workstation-list-policies (wlp) [options] (workstation object path)`

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-view-all-policies-status (wvaps) [options] (workstation path)`

Displays the advanced deployment status of policies assigned to a workstation. Accepts the following options:

--statusfilter=[status type][...] - Filter 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-refresh (wrf) [options] (workstation object path) [...]`

Refreshes the ZENworks Adaptive Agent in the workstations. Accepts the following options:

-f, --folder=[workstation folder] - 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-wakeup (ww) [options] (workstation object path) [...]`

Wakes up a workstation using Wake-On-LAN. Accepts the following options:

-f, --folder=[workstation folder] - Path of a workstation folder relative to /Devices/ Workstations. If 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.

ZENworks Server Commands

These commands are used to manage servers that have ZENworks 10 Configuration Management installed. The ZENworks Server commands begin with zenserver- in the long form, or with the letters zs in the short form.

`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.

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.

Global Options

The following options can be applied to any zman command:

-h, --help, ?, -?, /?

Provides help at various levels. Refer GUIDE TO USAGE section for more details on usage of help.

--usage

Display the syntax of the command.

--terse

Display terse output.

--debug

Display debugging output.

-V, --verbose

Enable verbose output.

--quiet

Quiet output, print only error messages.

-U, --User=<username>

Provide a user name. If not provided, you are prompted.

-P, --Password=<password>

Specify a password. If not provided, you are prompted.

---host=<host>

Specify the host name or IP address to connect to (default: localhost).

--port=<port>

Specify the port that the server is listening on (default: 443).

--cleartext

> Disable SSL for debugging purposes. If port is not set to the clear text port, it connects to port 80 by default.

-R, --Redirect=<file path>

> Redirect 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.

Files

zman-config.properties

This is a property file used to configure zman. The various options available are:

RENDERER_SCROLLSIZE: This controls the output of a list command globally. Results are fetched 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 Window 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

A sample content of the file would be 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 commands, `admin-store-credential` and `admin-clear-credential` should be used to add administrator credentials to each command. Refer the SECURITY section for guidelines on using administrator credentials.

GLOBAL_OPTIONS_HELP: Global options for the commands are displayed in every command help. To turn of the display of global options, set GLOBAL_OPTIONS_HELP=false.

DEBUG_LEVEL: Controls the amount 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, output of the command and more debug messages.

Security

Running of each command requires administrator authorization. To avoid entering administrator credentials for every command, use the command admin-store-credential to store the credentials. The credentials are encrypted and stored in the home folder of the operating system user running the

command. The credentials will only apply to that logged on user. It is safer to clear the information, by running admin-clear-credential command after execution of all the commands is done.

IMPORTANT: Though command line options for entering user name and password exist, avoid using them. In Linux, they will show up in the ps listing and pressing of up or down arrows key would show the previous commands entered in both Windows and Linux. Either enter the credentials when prompted or use the admin-store-credential command.

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 parse 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 - None of the options are entered when at least one is required. For example, to modify rights, you must specify at least the Assign 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

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 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.

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 of the content file to the temporary directory was cancelled because a file by that name already exists.

78 - Creation of a bundle or a policy with file content is being attempted on a non 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 Commands:

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 Commands:

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.

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 2005-2008, 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 the Novell® ZENworks® 10 Configuration Management.

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 install 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 (in the Common Tasks, click *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`.

To install the zeninfocollect utility on a Linux managed device having 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 (in the Common Tasks, click *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

Usage Options:

--output-dir

> Specifies the directory to place 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.

-o, --no-config

> Do not collect configuration data.

-d, --no-hardware

> Do not collect hardware data.

-l, --no-logs

> Do not collect logs.

-p, --no-packages

> Do not collect package data. Package data includes all version information for packages in the ZENworks 10 Configuration Management package repository.

-s, --no-server

> Do not collect server data.

Authors

Copyright 2005-2008, Novell, Inc (http://www.novell.com). All rights reserved.

zac(1)

Name

zac - The command line management interface for the Novell ZENworks Adaptive Agent.

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.

Commands

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"`

/h or --help

Display information about the commands.

add-reg-key (ark) <registration key>

Register the device 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`

bundle-install (bin) <bundle display name>

Install 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>

Launch 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
```

bundle-list (bl)

Display the list of bundles assigned to the device and the logged in user.

Example:

```
zac bl
```

bundle-props (bp) <bundle display name>

Display 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-remove (brm) <bundle display name>

Uninstall the specified bundle. Use the `bundle-list` command to get a list of installed bundles and their display names.

Example:

```
zac brm bundle1
```

bundle-uninstall (bu) <bundle display name>

Uninstall 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>

Verify 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
```

cache-clear (cc)

Clear 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
```

dump-prop-pages (dpp) <target directory>

Output the HTML pages displayed in the ZENworks icon's property pages to files in the specified tanoverget directory.

Example:

```
zac dpp c:\temp
```

get-settings (gs) <registration key>

List the settings associated with the specified registration key.

Example:

```
zac gs key1
```

logger

Change or display the logger configuration for the ZENworks Adaptive Agent.

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 current managed logging level.

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-export (pe) <policy type>

Export the settings for a Browser Bookmarks policy to an XML file or a Windows Group policy to a zip file.

The `policy type` must be one of the following types: browserbookmarkspolicy or grouppolicy.

Usage for Browser Bookmarks policy:

```
zac pe browserbookmarkspolicy <export XML filename> <Favorites folder to export>
```

Example for Browser Bookmarks policy:

```
zac pe browserbookmarkspolicy c:\financial.xml Financial
```

Usage for Windows Group policy:

```
zac pe grouppolicy <options> <export ZIP filename>
```

where `options` are:

`all` - Exports both computer and user configuration settings.
`computer (cmp)` - Exports computer configuration settings only.
`user (usr)` - Exports user configuration settings only.

active-directory (ad) [*Active Directory group policy path*] - Exports Active Directory group policy; requires you to specify Active Directory group policy path.

Example for Windows Group policy:

```
zac pe grouppolicy all c:\policy.zip
```

policy-list (pl)

List 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)

Apply all of the policies assigned to the device and user.

Example:

```
zac pr
```

refresh (ref) [general | partial | bundle <bundle display name>]

Initiate a general refresh to refresh all bundles, policies, registration, and configuration settings. Initiate a partial refresh to refresh all policies, registration, and configuration settings. Initiate a bundle refresh to refresh a specific bundle.

Examples:

```
zac ref general
zac ref bundle bundle1
```

register (reg) [-k <key>] [-u <username> -p <password>] <ZENworks Server address:port>

Register 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.

set-proxy (sp) [options] <IP address:port> [username] [password]

Specify 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.

unregister (unr) [-f] [-u <username> -p <password>]

> Remove the device's registration from the Management Zone. There is one option:
>
> 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 only necessary if the device object has already been deleted from the ZENworks database or if the device cannot connect to the database.

wake-cdp (cdp) [replicate | cleanup]

> Wake 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`

winproxy-refresh (wpr)

> Query the Management Zone for proxy work assigned to this device.
>
> Example:
>
> `zac wpr`

zone-config (zc) [-l]

> Display 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`

Authors

Copyright 2008, Novell, Inc. All rights reserved. http://www.novell.com

Troubleshooting ZENworks Command Line Utilities

2

The following sections provide solutions to the problems you might encounter while working with Novell® ZENworks® 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 67
- "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 67
- "Unable to save the report on SLES 10 SP1 by using the zman rpgn command" on page 67

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 10 Configuration Management; zman.

Possible Cause: zman requires the ZENworks administrator username and password for executing each command. If the credentials are not provided along with the command, then you are prompted to enter the same. But if the output of the command is redirected to file, then 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 the redirection operator such as > or >>.

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 10 Configuration Management, 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 which 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 10 Configuration Management; zman.

Action: None.

Guidelines for Working with Zman

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

Follow these guidelines as you work with the zman command line utility of Novell® ZENworks® 10 Configuration Management:

* 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, then you must open the file in an editor using the UTF-8 encoding. If you make any changes to 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 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 DOS prompt is cp437 or cp850 where as the rest of the Windows Operating System uses cp1252.

  To redirect the output to a file to 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 need not perform 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. If you want zman to use a specific language, do the following:

  * On Windows, specify the appropriate user language and file encoding as a value for the property JVM_STARTUP_OPTIONS 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}" .....`