

InterWorx Command-Line Interface Guide

by InterWorx LLC

Contents

1	CLI Quick Overview	2
1.1	Overview of the CLI Utility	2
2	Using the CLI Utility	4
2.1	Interactive Usage	4
2.2	Non-Interactive Usage	6
2.3	Summary	8

Chapter 1

CLI Quick Overview

The InterWorx Control Panel offers a command-line utility allowing server administrators to perform InterWorx functions without having to log in to the web interface. This utility can also be used in shell scripts to perform automated batch operations.

The CLI utility is part of the default InterWorx repository that is installed to your system when you first install InterWorx. You can use yum to install the utility by running the following command as root:

```
[root@server ~]# yum install interworx -cli
```

1.1 Overview of the CLI Utility

The CLI utility is comprised of two commands: 'nodeworx' and 'siteworx'. The 'nodeworx' command is used to perform administrator and reseller functions that you would normally access in the NodeWorx control panel; the 'siteworx' command is used to perform SiteWorx user functions on specific SiteWorx accounts that are normally accessed on the SiteWorx control panel.

For example, adding a new reseller would be done by using the 'nodeworx' command and adding a new e-mail alias for a SiteWorx user would be done by using the 'siteworx' command.

Both commands allow you to use the -h option to get a usage summary.

1.1.1 CLI Authentication

In order to use these commands, you must supply authentication. The basic way to authenticate is to simply use either the nodeworx or siteworx command and you will be prompted for credentials.

```
[root@server ~]# nodeworx
+-----+
| NodeWorx Login |
+-----+
E-mail : admin@server.com
Password [] :
```

```
[root@server ~]# siteworx
+-----+
| SiteWorx Login |
+-----+
E-mail : client@siteworx-example.com
```

```
Password [] : XXXXXXXXXXXX
Domain : siteworx-example.com
```

Notice that the SiteWorx login also requires that you enter the domain of the SiteWorx account in question, just like logging in via the web interface.

1.1.2 Unix User-Based Authentication

Since it would be silly to require someone who already had root access to the server to have to re-login again, it is possible to authenticate based on the Unix user. The root user may log into the master NodeWorx user by using:

```
[root@server ~]# nodeworx -u
```

This is also helpful when writing scripts because you will be able to skip the authentication prompt. With InterWorx version 4.6.0 and later, the root user may also login to any SiteWorx account with:

```
[root@server ~]# siteworx -u --login_domain=siteworx-example.com
```

SiteWorx users with shell access can also use the siteworx CLI utility and authenticate by logging in to a shell and entering the following command:

```
[siteworx@server ~]$ siteworx -u
```

That's it!

Chapter 2

Using the CLI Utility

2.1 Interactive Usage

Below you will find the list of available Controllers. Each controller is essentially a section of InterWorx. As you can see, they are exactly the same as the menu items in the NodeWorx web interface. Similarly for the siteworx command, the controllers listed are the same as the menu items in the SiteWorx web interface. You can select which controller you want to access by entering its menu number in the prompt.

```
[root@server ~]# nodeworx -u
+-----+
| Additional input is required |
+-----+
(1) Login / Logout
(2) NodeWorx Home
(3) NodeWorx: User Accounts
(4) NodeWorx: Languages
(5) NodeWorx: Themes
(6) NodeWorx: API Key
(7) NodeWorx: Plugins
(8) SiteWorx: Accounts
(9) SiteWorx: Shell Users
(10) SiteWorx: Packages
(11) SiteWorx: Backup / Restore
(12) SiteWorx: Import
(13) Simple Scripts
(14) Softaculous
(15) Resellers: Accounts
(16) Resellers: Packages
(17) Resellers: Bandwidth History
(18) System Services: Web Server
(19) System Services: FTP Server
(20) System Services: SSH Server
(21) Mail Server: MTA Settings
(22) Mail Server: SMTP Settings
(23) Mail Server: SMTP2 Settings
(24) Mail Server: Outbound SMTP Settings
```

```

(25) Mail Server: MDA Settings
(26) Mail Server: POP3 Settings
(27) Mail Server: Secure POP3 Settings
(28) Mail Server: IMAP Settings
(29) Mail Server: Secure IMAP Settings
(30) Mail Server: SPAM Filtering
(31) Mail Server: Virus Filtering
(32) Mail Server: Mail Queue
(33) MySQL Server: Overview
(34) MySQL Server: PhpMyAdmin
(35) DNS Server: Overview
(36) DNS Server: Zones
(37) DNS Server: Records
(38) DNS Server: Synchronization
(39) NFS Server: Overview
(40) NFS Server: Exports
(41) NFS Server: Mounts
(42) Server: System Health
(43) Server: CRON
(44) Server: IP Management
(45) Server: Firewall
(46) Server: SSL Certificates
(47) Server: Logs
(48) Server: Settings
(49) Server: Software Updates
(50) Server: System Graphs
(51) Clustering: Setup
(52) Clustering: Nodes
(53) Clustering: Load Balancing
(54) Clustering: Node Status
Controller [none] :

```

Let's say you just want to see what you can do with the "SiteWorx: Accounts Controller" so you enter '8'. Every controller has Actions associated with it, as you can see below.

```

Controller [none] : 8
+-----+
| Choose an Action |
+-----+
(1) toggleFavorite * Toggle a "favorite" SiteWorx account
(2) index * List Siteworx accounts
(3) siteworxLogin * Login to a given SiteWorx account
(4) deleteAccountOnNode * Delete a SiteWorx Account on this Cluster Node
(5) delete * Delete a SiteWorx Account
(6) bulkdeactivate * Bulk deactivate SiteWorx accounts
(7) bulkdelete * Bulk delete SiteWorx accounts
(8) bulkactivate * Bulk activate SiteWorx Accounts
(9) add * Add a SiteWorx account

```

```

(10) suspend * Suspend a SiteWorx account
(11) unsuspend * Unsuspend a SiteWorx account
(12) edit * Edit a SiteWorx account
(13) listMasterDomains * List master domains
(14) listFreeIps * List available ip addresses
(15) listDedicatedFreeIps * List available dedicated ip addresses
(16) listSharedFreeIps * List available shared ip addresses
(17) getSiteworxAccountDetails * Get siteworx account details
(18) querySiteworxAccountDetails * Get siteworx account details
(19) listBandwidthAndStorage * Lists bandwidth and storage usage for the current b
(20) listSiteworxAccounts * List SiteWorx accounts (id, status, domain)
(21) querySiteworxAccounts * Query Siteworx Accounts
(22) listNextUidAndGid * List the next available user and group ids
(23) syncVirtualHosts * Synchronize InterWorx and Apache virtual host blocks
(24) syncServerAlias * Synchronize server aliases with InterWorx database
(25) isARecorded * Checks if the current action is a recorded action or not
Action [none] :

```

These are what allow you to select the specific operation to perform. Say you wanted to add a new SiteWorx account - you enter '9' for the add action. You are then prompted for the necessary information to perform this action.

```

Action [none] : 9
2 +-----+
3 | SiteWorx Account Management
4 +-----+
5 Domain Name : usage-example.com
6 IP Address : 10.10.0.5
...

```

2.2 Non-Interactive Usage

While the interactive mode of the CLI utility may be useful, ultimately you probably want to use the CLI utility in scripts, which you can't do in interactive mode. In order to do this, you first need to figure out what the proper arguments are, and this is where the `-h` usage option becomes a great resource because it documents all the controllers, actions, and arguments you need to use to perform InterWorx tasks.

Let's say for example we want to learn how to add an FTP account to a SiteWorx account non-interactively. In order to do this, we need to authenticate just like we did above, but also add the `-h` option to the command line, like so:

```

[root@server ~]# siteworx -u --login_domain=example.com -h
Usage: siteworx.php [ options ]
--help|-h Display this help and exit
--session-id <string> Session ID for continuing a session
--user-auth|-u Authenticate by unixuser
--interactive|-i Render in interactive mode with a session
--non-interactive|-n Render in non-interactive mode for programming
--verbose|-v Outputs more verbose information in non-interactive mode

```

```

--output-style|-o <word> default|json
--controller|-c <string> Index|Overview|Email|EmailBox|EmailAlias|
    EmailGroup|EmailAutorespond|DomainsSlave|DomainsPointer|DomainsSub|
    Ftp|FtpSessions|MysqlDb|MysqlUser|MysqlPerms|Htaccess|Users|
    HttpdLogs|EmailRemotesetup|EmailFilters|EmailSpamprefs|EmailSpf|
    EmailDomainkeys|Prefs

```

Notice the bottom has ALL of the SiteWorx controllers listed. So, we specify the controller we want with `-c Ftp`, as you can see from the usage text. Yet, we still don't know what action we want to use. The great thing about the CLI utility is with the `-h` option and the `-c` controller option, the CLI utility will change the usage text to show you what actions are available:

```

[root@server ~]# siteworx -u --login_domain=example.com -h -c Ftp
action : This input is required

Usage: siteworx.php Ftp [ options ]
--help|-h Display this help and exit
--session-id <string> Session ID for continuing a session
--user-auth|-u Authenticate by unixuser
--interactive|-i Render in interactive mode with a session
--non-interactive|-n Render in non-interactive mode for programming
--verbose|-v Outputs more verbose information in non-interactive mode
--output-style|-o <word> default|json
--controller|-c <string> Index|Overview|Email|EmailBox|EmailAlias|
    EmailGroup|EmailAutorespond|DomainsSlave|DomainsPointer|DomainsSub|
    Ftp|FtpSessions|MysqlDb|MysqlUser|MysqlPerms|Htaccess|Users|
    HttpdLogs|EmailRemotesetup|EmailFilters|EmailSpamprefs|EmailSpf|
    EmailDomainkeys|Prefs
--action|-a <string> index|add|edit|delete|list|listDeletable

```

Notice that now after the list of controller options, the list of actions you can do with the FTP controller is listed. If you specify the 'add' action, the `-h` option again will change to show you what arguments you need to add an FTP account.

```

[root@server ~]# siteworx -u --login_domain=example.com -h -c Ftp -a add
Usage: siteworx.php Ftp add [ options ]
--help|-h Display this help and exit
--session-id <string> Session ID for continuing a session
--user-auth|-u Authenticate by unixuser
--interactive|-i Render in interactive mode with a session
--non-interactive|-n Render in non-interactive mode for programming
--verbose|-v Outputs more verbose information in non-interactive mode
--output-style|-o <word> default|json
--user <string>
--password <string>
--confirm_password <string>
--homedir <string>

```

Now the controller and action documentation has been replaced with the arguments you pass to the action (`-user`, `-password`, `-confirm_password`, `-homedir`) at the bottom. To construct the final command which

will add the FTP account, we need to remove the `-h` option because we don't need documentation anymore and we add the `-n` non-interactive option to make the CLI utility not prompt us for anything.

```
[root@server ~]# siteworx -u -n --login_domain=example.com -c Ftp -a add --user ftp
```

This will silently, non-interactively, add a SiteWorx FTP account. Notice that for `-password` arguments, you need the `-confirm_password` argument too. If you want to see the result of your non-interactive command, you can optionally add a `-v` which enables verbose mode. You can follow this procedure to figure out the proper controller/action/arguments combination to perform any InterWorx task you desire.

2.3 Summary

1. Install with `yum install interworx-cli`
2. Use the `-h` option for documentation.
3. Use the `-n` option for non-interactive mode.
4. Use the `-v` option for verbose mode. Good for non-interactive mode if you want see a success message printed to the screen.

2.3.1 NodeWorx CLI Synopsis

```
nodeworx [-u] [-v] [-n |-i] [-h] [ -c CONTROLLER ] [ -a ACTION ] [ ARGUMENTS ]
```

2.3.2 SiteWorx CLI Synopsis

```
siteworx [-u] [--login_domain DOMAIN] [-v] [-n |-i] [-h] [ -c CONTROLLER ] [ -a AC
```