Contents

- 1 About the Document
- 2 Pisi CLI Operations
  - 2.1 Repository Operations
    - 2.1.1 Adding Repositories
    - 2.1.2 Removing Repositories
    - 2.1.3 Listing Repositories
    - 2.1.4 Updating Repositories
    - 2.1.5 Listing Components
    - 2.1.6 Creating Repository Index
  - 2.2 Package Operations
    - 2.2.1 Installing Packages
    - 2.2.2 Installing just one package without installing all the upgradable packages available
    - 2.2.3 Removing packages
    - 2.2.4 Upgrading Packages
    - 2.2.5 Excluding Packages from upgrade
    - 2.2.6 Updating just one package without updating the others
    - 2.2.7 Getting Package Information
    - 2.2.8 Building Packages from Source
    - 2.2.9 Building Packages from Source Repository
    - 2.2.10 Listing Packages
    - 2.2.11 Listing Upgrades
    - 2.2.12 Listing Installed Packages
    - 2.2.13 Listing Pending Packages
    - 2.2.14 Listing Source Packages
    - 2.2.15 Configuring Pending Packages
    - 2.2.16 Searching Packages
    - 2.2.17 Checking Packages
  - 2.3 Other Operations
    - 2.3.1 Rebuilding PiSi Database
    - 2.3.2 Deleting Files in Cache
    - 2.3.3 Searching Files
    - 2.3.4 Getting Package Relations
    - 2.3.5 Removing Locks
    - 2.3.6 Displaying history
    - 2.3.7 Getting Help

About the Document

This document is intended to provide an overview of pisi usage on the command line (pisi-cli).

Pisi CLI Operations

Repository Operations

Adding Repositories

add-repo command with *repo-name* and *repo-address* parameters is used in order to add repositories.

```
pisi add-repo repo-address http://repo.address.com/repo/directory/pisi-index.xml
```

Possible values for the "http://en.pardus-wiki.orgrepo-address"http://en.pardus-wiki.org parameter:

- Local addresses i.e. /home/example/pisi/repo/pisi-index.xml
- Web addresses i.e. http://example.com/pisi/repo/pisi-index.xml

Note: Default name for repository index file is *pisi-index.xml*, but compressing it as *pisi-index.xml.bz2* can also be used in order to prevent high bandwidth usage.

Removing Repositories

remove-repo command with *repo-name* parameter is used in order to remove repositories.

```
pisi remove-repo repo-name
```

Listing Repositories

list-repo command is used in order to list which repositories are available.

```
pisi list-repo
```

Updating Repositories

update-repo command is used in order to update all available repositories.

```
pisi update-repo
```

One or more *repo-name* parameters can be used in order to update proper repositories.

```
pisi update-repo repo-name
```

Listing Components

list-components command is used in order to list all components in available repositories.

```
pisi list-components
```

One or more *repo-name* parameters can be used in order to list components in proper repositories.

```
pisi list-components repo-name
```

Creating Repository Index

index command in the repo directory is used in order to create the repository index (pisi-index.xml).

```
pisi index
```

Package Operations

Installing Packages

install command with *package-name* parameter is used in order to install packages. Multiple packages can be installed by separating the package names by a space.

```
pisi install package-name
```

```
pisi install package-name1 package-name2
```

Installing just one package without installing all the upgradable packages available

```
pisi it package-name --ignore-safety
```

Removing packages

remove command with *package-name* parameter is used in order to remove packages. Multiple packages can be removed by separating the package names by a space.

```
pisi remove package-name
```

```
pisi remove package-name1 package-name2
```

Upgrading Packages

upgrade command with *package-name* parameter is used in order to upgraded packages. Multiple packages can be upgraded by separating the package names by a space.

```
pisi upgrade package-name
```

```
pisi upgrade package-name1 package-name2
```

All available packages can be upgraded by using *upgrade* command without any parameters.

```
pisi upgrade
```

Excluding Packages from upgrade

To update all the updatable packages except *firefox* :

```
pisi upgrade -x firefox
```

Similarly, if you don't want to update the packages that are part of the *desktop.kde* system component, do :

```
pisi upgrade -x desktop.kde
```

Updating just one package without updating the others

```
pisi upgrade package-name --ignore-safety
```

Getting Package Information

info command with *package-name* parameter is used in order to get information about a package. Getting information for multiple packages can be achieved by separating the packages names by a space.

```
pisi info package-name
```

```
pisi info package-name1 package-name2
```

Building Packages from Source

build command with *package-address* parameter is used in order to build packages from source.

```
pisi build /home/example/pisi/repo/pisi-index.xml
```

Building Packages from Source Repository

emerge command with *package-name* parameter is used in order to build a package and all dependencies from the source repository.

```
pisi emerge package-name
```

Listing Packages

list-available command with *repo-name* parameter is used in order to list all packages in the proper repository.

```
pisi list-available repo-name
```

list-available command is used in order to list all packages in the all repositories.

```
pisi list-available
```

Listing Upgrades

list-upgrades command is used in order to list upgrades in all repositories.

```
pisi list-upgrades
```

list-upgrades command with *repo-name* parameter is used in order to list upgrades in a proper repository.

```
pisi list-upgrades repo-name
```

Listing Installed Packages

list-installed command is used in order to list installed packages.

```
pisi list-installed
```

list-installed command with *repo-name* parameter is used in order to list installed packages from a proper repository.

```
pisi list-installed repo-name
```

Listing Pending Packages

list-pending command is used for listing pending package operations.

```
pisi list-pending
```

Listing Source Packages

list-sources command is used for listing available source packages

Configuring Pending Packages

configure-pending command is be used in order to configure packages which are not already configured by COMAR.

```
pisi configure-pending
```

Searching Packages

search command with one or more "http://en.pardus-wiki.orgsearch-key"http://en.pardus-wiki.org parameters is used in order to search packages.

```
pisi search search-key1 search-key2
```

Checking Packages

check command with *package-name* parameter is used for checking if the package is installed correctly.

```
pisi check package-name
```

Other Operations

Rebuilding PiSi Database

rebuild-db command is used in order to rebuild pisi package database.

```
pisi rebuild-db
```

Deleting Files in Cache

delete-cache command is used in order to delete files(packages, source codes etc.) stored in cache.

```
pisi delete-cache
```

Searching Files

search-file command with *file-name* parameter is used in order to find which package does a file belong to.

```
pisi search-file /path/file
```

Getting Package Relations

graph command with *package-name* parameter is used for getting package relations via graphviz.

```
pisi graph package-name
```

Removing Locks

clean command is used in order to delete outdated locks.

```
pisi clean
```

Displaying history

history command displays the history of installations/deinstallations/updates of packages.

```
pisi history
```

Getting Help

help command is used in order to get help about pisi-cli.

```
pisi help
```

help command with *command-name* parameter can be used for getting help about a specific command.

```
pisi help command-name
```