Name

xdcpkg - XDC package list command

Synopsis

```
xdcpkg [-a] [-d] [-l[:fmt]] [-m[c]] [-p path] [-r] [-s]
[dir …]
```

Description

The `xdcpkg` command is used to find and display information about packages. By default, `xdcpkg` finds only packages that can be built (i.e., packages with both `package.xdc` and `package.bld` files).

Options

`-a` display all packages, not just the buildable ones,

`-d` treat each directory as a package directory and (recursively) find the package’s pre-requisites and display them.

`-l[:fmt]` display all package information, not just each package’s base directory. If the optional ‘:’ is used, the remainder of the option is taken to be a “format” string that controls the contents and format of the output. The format string is similar to a printf format string; each character (except %) in the string is written to stdout, “output specifiers” are introduced by the % character, the special characters ‘%’, ‘\’, tab, new line, and carriage return are output by the character pairs '\%', '\\', '\t', '\n', or '\r' respectively. The following "output specifiers are supported":

`%b` base directory of package

`%c` release build counter

`%d` release date in milliseconds since midnight (GMT) 1/1/1970

`%k` package's compatibility key (as specified by the package’s `package.xdc` file)

`%l` package’s release label

`%n` package’s name

`%r` package’s release name

`%t` top directory containing package (i.e., the package’s repository)

`%v` a “version” number created by appending the package's compatibility key with the build counter

`%V` a “version” number created by appending the package's compatibility key with the release date in milliseconds

If `fmt` is not specified, print package names followed by their base directory, release name, release label, version, and release date using the following format:

```
%n;%b;%r;%l;%v;%d
```

`-m[c]` output make dependencies in the form `a:b` where `a` and `b` are package base directories. If the character `c` is specified, the ':' character is replaced by `c`.

`-p path` the path to use as the *complete* package path (not just `$XDCPATH`)

`-r` treat each directory as a repository and display *only* those packages that are “rooted” in this repository. Without this option, each directory is recursively searched for any package in any repository at or below the starting directory

`-s` shallow package search; don't recurse when looking for packages. If `-d` is also specified only display immediate requirements. This option can't be used with `-r`.

`-v` verbose output; display error messages for any failure.

Examples

Finding Packages

The following command will output all directories, starting from the current working directory, containing a buildable package.

```
xdcpkg .
```

To find all packages (buildable or not) starting from the current working directory:

```
xdcpkg -a .
```

To find all buildable prerequisite packages for the package whose base is the current working directory:

```
xdcpkg -d .
```

To find an appropriate build order for all buildable prerequisite packages of the package whose base is `./`:

```
xdcpkg -m" " -d . | tsort
```

To find all prerequisite packages for the package whose base is the current working directory:

```
xdcpkg -a -d .
```

To find all packages whose repository is `./src`:

```
xdcpkg -a -r ./src
```

Getting Package Information

The following command will display just the names of all packages located in (or below) the directories `./src` and `c:/ti/packages`:

```
xdcpkg –a –l:%n ./src c:/ti/packages
```

To output the names and a unique version identifier (between []'s) for all packages installed in the repository `c:/ti/packages`:

```
xdcpkg -r -a -l:"%n[%V, %r]" c:/ti/packages
```

Environment Variables

In the event that the `–p` flag is not specified, `xdcpkg` constructs a package path from the environment variables `XDCROOT` and `XDCPATH` as follows:

```
$XDCPATH;$XDCROOT/packages;^
```

where,

`XDCPATH` a string of ';' separated directories that contain packages. This path is used to locate packages. Recall that the '`^`' character in the `XDCPATH` definition refers to the current package's repository.

`XDCROOT` This variable names an installation directory of the XDC tools.

Exit Status

The following exit values are returned:

0 Successful completion.

>0 An error occurred.