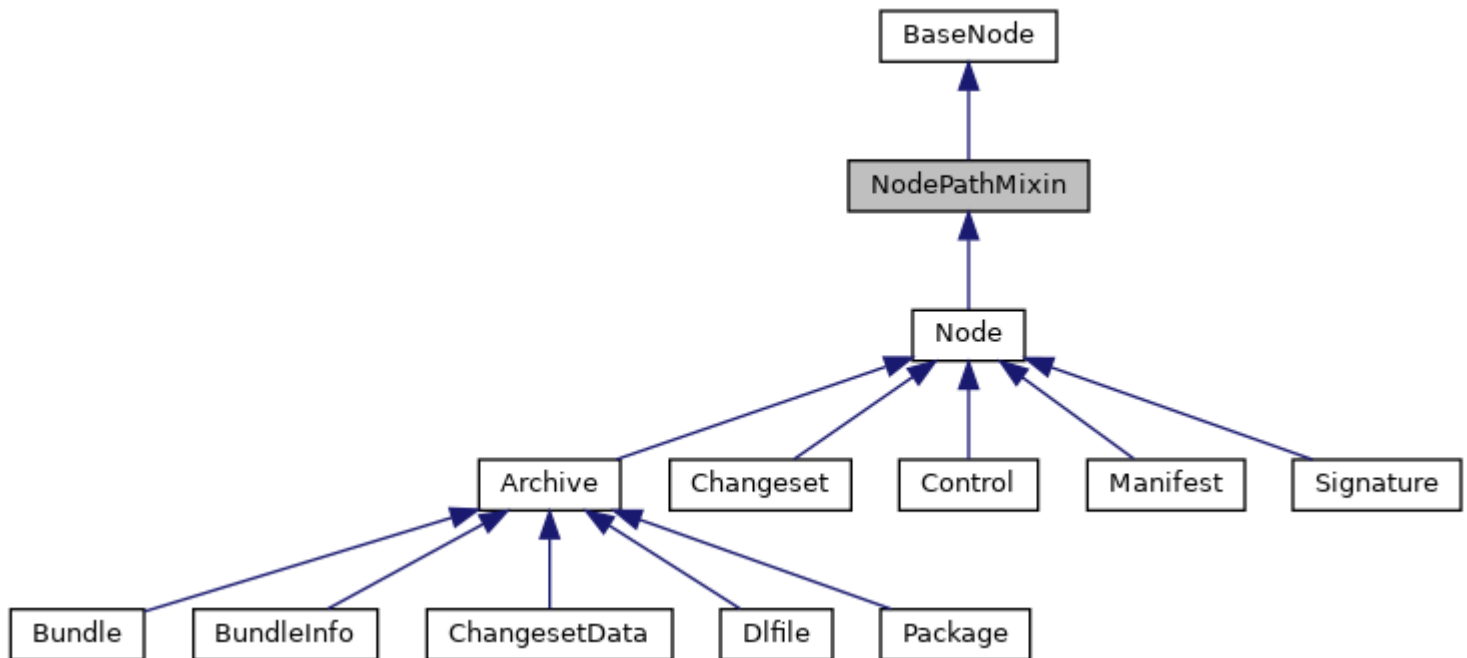


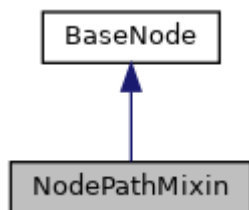
NodePathMixin Class Reference

Inheritance diagram for NodePathMixin:



[\[legend\]](#)

Collaboration diagram for NodePathMixin:



[\[legend\]](#)

Public Member Functions

def [get_path_nodes](#) (self, include_root=False, within_archive=False)

str [`get\_path`](#) (self, include_root=False, within_archive=False)
 Retrieve path of this node within the archive or withing whole tree of archives Separator '/' is used within archives and for recursion into archives. [More...](#)

int [`get\_depth`](#) (self, within_archive=False)
 Retrieve depth of node within the archive or within whole tree of archives. [More...](#)

def [`find`](#) (self, str node_path)
 Find node according to node path provided. [More...](#)

def [`find\_re`](#) (self, str node_path_re)
 Find node according to regular expression node path provided Search will look for regex matching node name match along the path. [More...](#)

bool [`has\_ancestor`](#) (self, ancestor)
 Find if node has another node as ancestor. [More...](#)

► Public Member Functions inherited from [BaseNode](#)

def [`\_\_init\_\_`](#) (self, [`name`](#)=None, [`parent`](#)=None, [`tarinfo`](#)=None)

str [`get\_type\_str`](#) (self)
 Returns node type as string Supported are: "Package", "Bundle", "Dlfile", "Signature", "Control", "Manifest", "Dir", "File", "Symlink". [More...](#)

bool [`is\_csd`](#) (self)
 Test if node is a CSD. [More...](#)

bool [`is\_inf`](#) (self)
 Test if node is an INF. [More...](#)

bool [`is\_manifest`](#) (self)
 Test if node is a manifest. [More...](#)

bool [`is\_changeset`](#) (self)
 Test if node is a changeset. [More...](#)

bool [`is\_archive`](#) (self)
 Test if node is an archive. [More...](#)

bool [`is\_dlfile`](#) (self)
 Test if node is a dlfile. [More...](#)

bool [`is\_bundle`](#) (self)
 Test if node is a bundle. [More...](#)

bool [`is\_package`](#) (self)
 Test if node is a package. [More...](#)

bool [`is\_signature`](#) (self)
 Test if node is a signature. [More...](#)

bool [`is\_control`](#) (self)
 Test if node is a control. [More...](#)

def [`mark\_modified`](#) (self)

def [`get\_parent\_archive`](#) (self)
 Retrieve parent archive node. [More...](#)

def [`get\_platform`](#) (self)

Retrieve platform for this node. [More...](#)

def [get_root_archive](#) (self)
 Retrieve root archive node Root archive node has no parent archive. [More...](#)

int [get_index](#) (self)
 Retrieve index of this node in the list of its parent's children. [More...](#)

def [get_signer](#) (self)

def [__str__](#) (self)

Additional Inherited Members

► Static Public Member Functions inherited from [BaseNode](#)

tarfile.TarInfo [create_tarinfo](#) (str tarinfo_name, entry_type)
 Tarinfo creation helper. [More...](#)

def [create_node](#) (tarinfo_name, entry_type)

► Data Fields inherited from [BaseNode](#)

[name](#)
 node name (str) [More...](#)

[parent](#)
 parent node ([BaseNode](#)) [More...](#)

[children](#)
 children of this node in order. [More...](#)

[tarinfo](#)
 Tarinfo object containing the type of archive entry, path, user, group, permission, etc. [More...](#)

[file_content](#)
 Object representing the content of the file. [More...](#)

[signature_nodes](#)
 Nodes holding signature of this node. [More...](#)

Member Function Documentation

[?](#) find()

```
def find (    self,
           str node_path
           )
```

Find node according to node path provided.

Search will look for exact node name match along the path. Example:
`dlfile.find("mybundle.tgz/mypackage.tgz/dir/file")`

Parameters

`node_path` node path

Returns

node (BaseNode) if found, None if not

[? find_re\(\)](#)

```
def find_re (    self,  
               str node_path_re  
             )
```

Find node according to regular expression node path provided Search will look for regex matching node name match along the path.

Only the leaf node is accepted to have multiple matches

Parameters

`node_path_re` node path regex

Returns

a list of nodes matching, None if none found

[? get_depth\(\)](#)

```
int get_depth ( self,  
                within_archive = False  
              )
```

Retrieve depth of node within the archive or within whole tree of archives.

Example for 'dlfile.tgz/bundle.tgz/package.tgz/dir/file':

- `within_archive=False`: -> 4
- `within_archive=True`: -> 2

Parameters

`within_archive` stop at first parent archive

Returns

depth of node

[? get_path\(\)](#)

```
str get_path ( self,  
              include_root = False,  
              within_archive = False  
            )
```

Retrieve path of this node within the archive or withing whole tree of archives Separator '/' is used within archives and for recursion into archives.

Example for 'dlfile.tgz/bundle.tgz/package.tgz/dir/file':

- include_root=False, within_archive=False -> 'bundle.tgz/package.tgz/dir/file'
- include_root=True, within_archive=False -> 'dlfile.tgz/bundle.tgz/package.tgz/dir/file'
- include_root=True/False, within_archive=True -> 'dir/file'

Parameters

include_root include root in path

within_archive stop at first parent archive

Returns

path of the node

[? get_path_nodes\(\)](#)

```
def get_path_nodes ( self,  
                    include_root = False,  
                    within_archive = False  
                  )
```

[? has_ancestor\(\)](#)

```
bool has_ancestor ( self,  
                   ancestor  
                 )
```

Find if node has another node as ancestor.

Parameters

ancestor ancestor (BaseNode) to look for

Returns

return True if ancestor is found

The documentation for this class was generated from the following file:

- `sys/import/packman/packmanlib/nodepath.py`