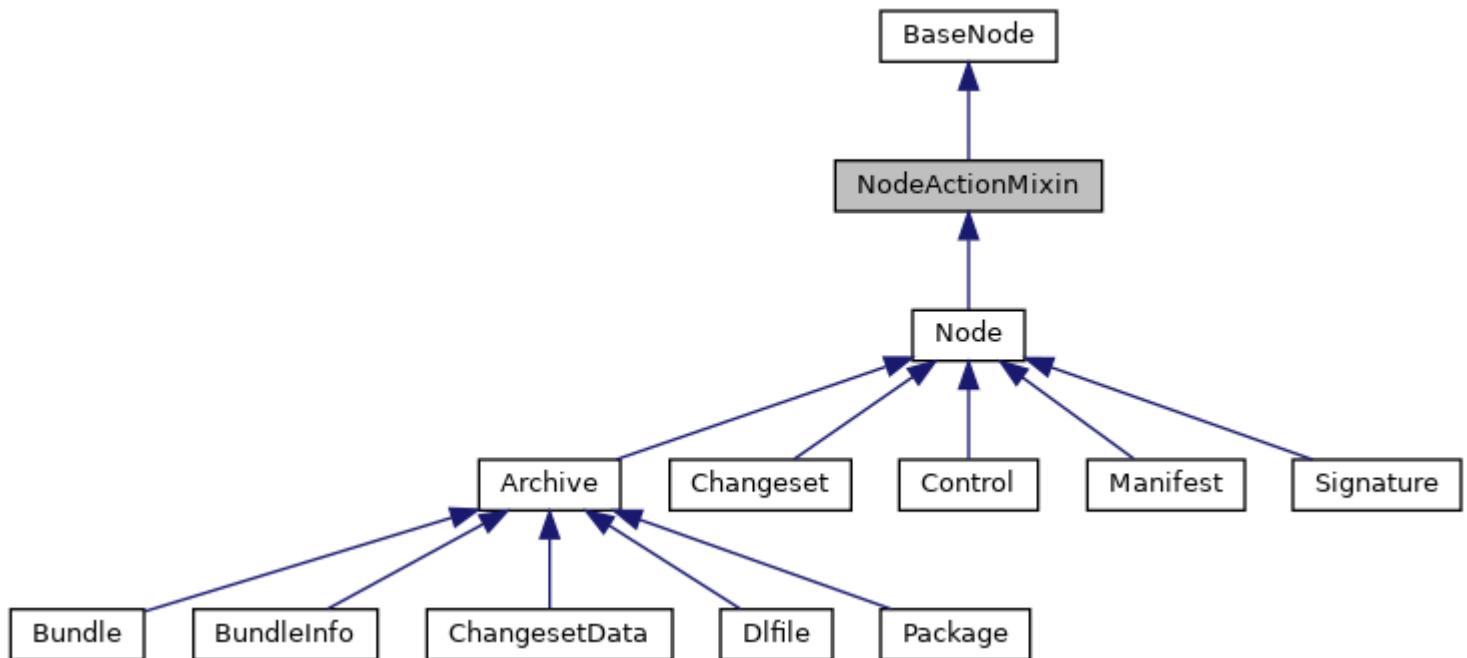


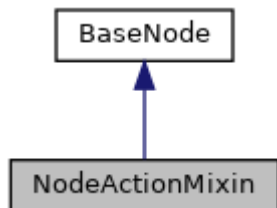
NodeActionMixin Class Reference

Inheritance diagram for NodeActionMixin:



[\[legend\]](#)

Collaboration diagram for NodeActionMixin:



[\[legend\]](#)

Public Member Functions

def [unlink](#) (self)

Remove this node from tree. [More...](#)

def [destroy](#) (self)
Remove and destroy this node and all its children. [More...](#)

def [add_child](#) (self, child_node, index=None)
Add a child node. [More...](#)

def [create_dir](#) (self, dir_name)
Add a child directory If same child name exists, it will be replaced. [More...](#)

def [create_file](#) (self, file_name, content=b'')
Add a child file. [More...](#)

def [set_content](#) (self, content)
Set the content of this file node. [More...](#)

def [set_mode](#) (self, mode)
Set the mode of this file/directory. [More...](#)

def [copy](#) (self)
Make a deep copy of this node and its content, children and attributes. [More...](#)

def [move](#) (self, [parent](#))
Move this node to another parent. [More...](#)

def [rename](#) (self, newname)
Rename this node and underlying tarinfo name. [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)

bool Test if node is a signature. [More...](#)
 [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)

Data Fields

[parent](#)
[children](#)
[file_content](#)
[name](#)
[signature_nodes](#)

► 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...](#)

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)
```

Member Function Documentation

[?](#) **add_child()**

```
def add_child ( self,  
               child_node,  
               index = None  
               )
```

Add a child node.

If same child name exists, it will be replaced

Parameters

child_node node to be added

index index where to insert node, None means append

Returns

added node or None if it could not be added

[?](#) **copy()**

```
def copy ( self )
```

Make a deep copy of this node and its content, children and attributes.

Returns

copied node

[?](#) **create_dir()**

```
def create_dir ( self,  
                dir_name  
                )
```

Add a child directory If same child name exists, it will be replaced.

Parameters

dir_name name of the directory

Returns

created node

[?](#) create_file()

```
def create_file ( self,  
                 file_name,  
                 content = b ' '  
                 )
```

Add a child file.

Parameters

file_name name of the file

content content of file

Returns

created node

[?](#) destroy()

```
def destroy ( self )
```

Remove and destroy this node and all its children.

Returns

removed node

[?](#) move()

```
def move ( self,  
          parent  
          )
```

Move this node to another parent.

Parameters

parent new parent node

Returns

this node

[**? rename\(\)**](#)

```
def rename ( self,  
            newname  
            )
```

Rename this node and underlying tarinfo name.

Parameters

newname new name for the node

Returns

this renamed node

[**? set_content\(\)**](#)

```
def set_content ( self,  
                 content  
                 )
```

Set the content of this file node.

Parameters

content content of file

Returns

this node

[**? set_mode\(\)**](#)

```
def set_mode ( self,  
              mode  
              )
```

Set the mode of this file/directory.

Parameters

mode octal mode value

Returns

this node

[?](#) unlink()

```
def unlink ( self )
```

Remove this node from tree.

Node and children are not destroyed.

Returns

removed node

Field Documentation

[?](#) children

children

[?](#) file_content

file_content

[?](#) name

name

[?](#) parent

parent

[?](#) signature_nodes

signature_nodes

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

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