OSE Workbench Core

Release 0.1.0a4

G Roques

Jul 13, 2020

OSECORE PACKAGE

Core library common to all Open Source Ecology (OSE) workbenches.

The OSE workbench core library aims to provide functionality in the following key areas:

Description	Package
Attaching parts to one another.	`osecore.app.attachment`
Manipulating "shapes" (e.g. vertexes, edges, faces, etc.).	`osecore.app.shape`
Generating cut-lists.	`osecore.gui.cut_list`
User selection in FreeCAD's GUI.	`osecore.gui.selection`

Explore the **osecore package** below for a complete reference of exposed functionality.

CHAPTER

ONE

OSECORE

OSE Core base package containing *app* and *gui*.

app is independent from *gui*, and thus does not know about `FreeCADGui`.

1.1 osecore.app

1.1.1 osecore.app.attachment

Exposes code for attaching objects to one another.

Name	Description
AttachmentError	Raise when unable to attach one object to another.

---

exception AttachmentError

Bases: `ValueError`

Raise when unable to attach one object to another.

1.1.2 osecore.app.shape

Exposes functions for the following topological data types or "shapes":

- compound
- compsolid
- solid
- shell
- *face*
- wire
- *edge*
- vertex

Shape is a generic term covering all of the above.

See Also: Topological data scripting

Name	Description
`move_parts`	Move parts based on placement, origin translation offset,
`place_shape`	Apply a placement to a given shape.
`place_shapes`	Apply a placement to a given list of shapes.

---

move_parts(*parts: List[Part.Shape]*, *placement: Base.Placement*, *origin_translation_offset: Base.Vector*, *reference_dimensions: List[float]*, *rotation: Base.Rotation = Rotation(0.0, 0.0, 0.0, 1.0)*) → None

Move parts based on placement, origin translation offset, reference dimensions, and optionally a rotation.

Parameters

- **parts** (`List[Part.Shape]`) – List of parts to move.
- **placement** (`Placement`) – A placement to apply to the given parts.
- **origin_translation_offset** (`Vector`) – Offset the parts to the origin.
- **reference_dimensions** (`Vector`) – Reference dimensions
- **rotation** (`Rotation, optional`) – Rotation, defaults to Rotation()

place_shape(*shape: Part.Shape*, *placement: Base.Placement*) → None

Apply a placement to a given shape.

See Also: https://www.freecadweb.org/api/db/d71/classPart_1_1TopoShapePy.html

Parameters

- **shape** (`Part.Shape`) – A shape.
- **placement** (`FreeCAD.Placement`) – Placement to apply to part

place_shapes(*shapes: List[Part.Shape]*, *placement: Base.Placement*) → None

Apply a placement to a given list of shapes.

See Also: https://www.freecadweb.org/api/db/d71/classPart_1_1TopoShapePy.html

Parameters

- **shapes** (`List[Part.Shape]`) – A list of shapes
- **placement** (`FreeCAD.Placement`) – Placement to apply to part

osecore.app.shape.edge

Name	Description
`find_edges_connected_to_vertex`	Find the edges connected to a vertex.
`is_edge_parallel_to_x_axis`	Check if the given edge is parallel to the X axis.
`is_edge_parallel_to_y_axis`	Check if the given edge is parallel to the Y axis.
`is_edge_parallel_to_z_axis`	Check if the given edge is parallel to the Z axis.

---

find_edges_connected_to_vertex(*vertex: Part.Vertex*, *edges: List[Part.Edge]*) → List[Part.Edge]

Find the edges connected to a vertex.

Parameters

- **vertex** (*Part.Vertex*) – A vertex to find the edges connected to.
- **edges** (*List[Part.Edge]*) – List of edges to search for the vertex.

Returns A new filtered list of edges that are connected to the vertex.

Return type List[Part.Edge]

is_edge_parallel_to_x_axis(*edge: Part.Edge*) → bool

Check if the given edge is parallel to the X axis.

Parameters **edge** (*Part.Edge*) – Edge to check if it's parallel to the X axis.

Returns `True` if parallel to the X axis, `False` otherwise.

Return type bool

is_edge_parallel_to_y_axis(*edge: Part.Edge*) → bool

Check if the given edge is parallel to the Y axis.

Parameters **edge** (*Part.Edge*) – Edge to check if it's parallel to the Y axis.

Returns `True` if parallel to the Y axis, `False` otherwise.

Return type bool

is_edge_parallel_to_z_axis(*edge: Part.Edge*) → bool

Check if the given edge is parallel to the Z axis.

Parameters **edge** (*Part.Edge*) – Edge to check if it's parallel to the Z axis.

Returns `True` if parallel to the Z axis, `False` otherwise.

Return type bool

osecore.app.shape.face

Name	Description
is_face_parallel_to_xy_plane	Check if the given face is parallel to the XY plane.
is_face_parallel_to_xz_plane	Check if the given face is parallel to the XZ plane.
is_face_parallel_to_yz_plane	Check if the given face is parallel to the YZ plane.
is_face_planar	Returns whether a face is planar or not.
make_face_from_vectors	Make a Face from list of vectors.

---

is_face_parallel_to_xy_plane(*face: Part.Face*) → bool

Check if the given face is parallel to the XY plane.

Parameters **face** (*Part.Face*) – A face.

Returns `True` if the face is parallel to the XY plane, `False` otherwise.

Return type bool

is_face_parallel_to_xz_plane(*face: Part.Face*) → bool

Check if the given face is parallel to the XZ plane.

Parameters **face** (*Part.Face*) – A face.

Returns `True` if the face is parallel to the XZ plane, `False` otherwise.

Return type bool

is_face_parallel_to_yz_plane(*face: Part.Face*) → bool

Check if the given face is parallel to the YZ plane.

Parameters **face** (*Part.Face*) – A face.

Returns `True` if the face is parallel to the YZ plane, `False` otherwise.

Return type bool

is_face_planar(*face: Part.Face*) → bool

Returns whether a face is planar or not.

Returns `False` for cylindrical faces like holes.

Parameters **face** (*Part.Face*) – A face.

Returns `True` if the face is planar, `False` otherwise.

Return type bool

make_face_from_vectors(*vectors: List[Base.Vector]*) → Part.Face

Make a Face from list of vectors.

See Part::TopoShapeFacePy Class Reference: https://www.freecadweb.org/api/d9/d35/classPart_1_1TopoShapeFacePy.html

Raises

- **ValueError** – When there's less than three vectors in a list.
- **ValueError** – When vectors don't form a closed wire.

Returns A face

Return type Part.Face

1.1.3 model

Name	Description
Model	Base class for models that encapsulate the data (i.e. topography and shape)

---

class Model(*obj*)

Bases: `object`

Base class for models that encapsulate the data (i.e. topography and shape) for a part, and is separate from the "view" or GUI representation.

onDocumentRestored(*fp*)

Executed after a document is restored, or a FeaturePython object is copied or duplicated.

Parameters **fp** (*Part::FeaturePython*) – Custom feature python object

1.1.4 three_dimensional_space_enums

A set of enumerations relating to three-dimensional space.

Name	Description
CoordinateAxis	A reference line in a three-dimensional cartesian coordinate system.
Plane	A plane in a three-dimensional cartesian coordinate system.
Side	A sides of a three-dimensional object available as one of FreeCAD's standard views.

---

class CoordinateAxis

Bases: `object`

A reference line in a three-dimensional cartesian coordinate system.

X = 'x'

Y = 'y'

Z = 'z'

class Plane

Bases: `object`

A plane in a three-dimensional cartesian coordinate system.

XY = 'xy'

XZ = 'xz'

YZ = 'yz'

class Side

Bases: `object`

A sides of a three-dimensional object available as one of FreeCAD's standard views.

See Also: Std View Menu

BOTTOM = 'bottom'

FRONT = 'front'

LEFT = 'left'

REAR = 'rear'

RIGHT = 'right'

TOP = 'top'

1.2 osecore.gui

Package for code relating to FreeCAD's GUI.

1.2.1 osecore.gui.cut_list

Package for code relating to creating a cut list in the GUI.

Name	Description
CutListItem	Represents an item in a cut-list.
create_copy_cut_list_to_clipboard_task_panel	Create a task panel to copy a cut-list to the user's clipboard.
create_save_cut_list_to_file_task_panel	Create a task panel to save a cut-list to the user's filesystem.

---

class CutListItem(*quantity: str*, *description: str*, *length: str*)

Bases: `tuple`

Represents an item in a cut-list.

Used for typing purposes only.

property description

Alias for field number 1

property length

Alias for field number 2

property quantity

Alias for field number 0

create_copy_cut_list_to_clipboard_task_panel(*cut_list_items: List[osecore.gui.cut_list._create_cut_list_task_panel.cut_list_item.C merge_cut_list_items_by_length: bool = False, note: Optional[str] = None*) → osecore.gui.cut_list._create_cut_list_task_panel.task_panel.copy_cut_lis

Create a task panel to copy a cut-list to the user's clipboard.

Simple Usage

```
cut_list = [
    {'quantity': '1', 'description': 'Foo', 'length': '3 in'},
    {'quantity': '1', 'description': 'Bar', 'length': '3 in'}]
panel = create_copy_cut_list_to_clipboard_task_panel(cut_list)
Gui.Control.showDialog(panel)
```

Merge Cut List Items by Length

```
cut_list = [
    {'quantity': '1', 'description': 'Foo', 'length': '3 in'},
    {'quantity': '1', 'description': 'Bar', 'length': '3 in'}]
panel = create_copy_cut_list_to_clipboard_task_panel(cut_list,
    merge_cut_list_items_by_length=True)
Gui.Control.showDialog(panel)
```

Parameters

- **cut_list_items** (*List[CutListItem]*) – A cut list.
- **merge_cut_list_items_by_length** (*bool, optional*) – Whether to merge cut-list items by length, defaults to `False`

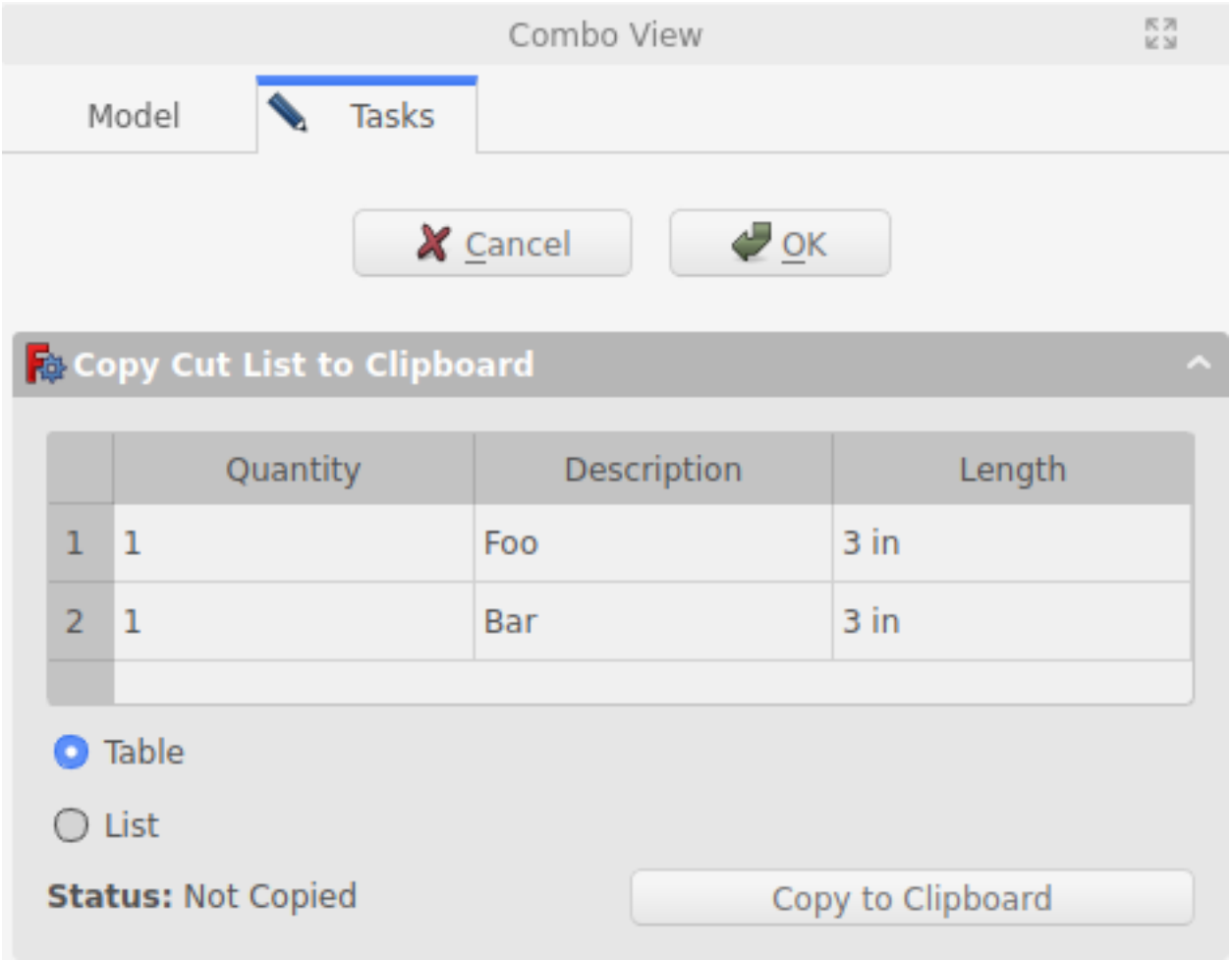

Fig. 1: Copy Cut List to Clipboard Task Panel

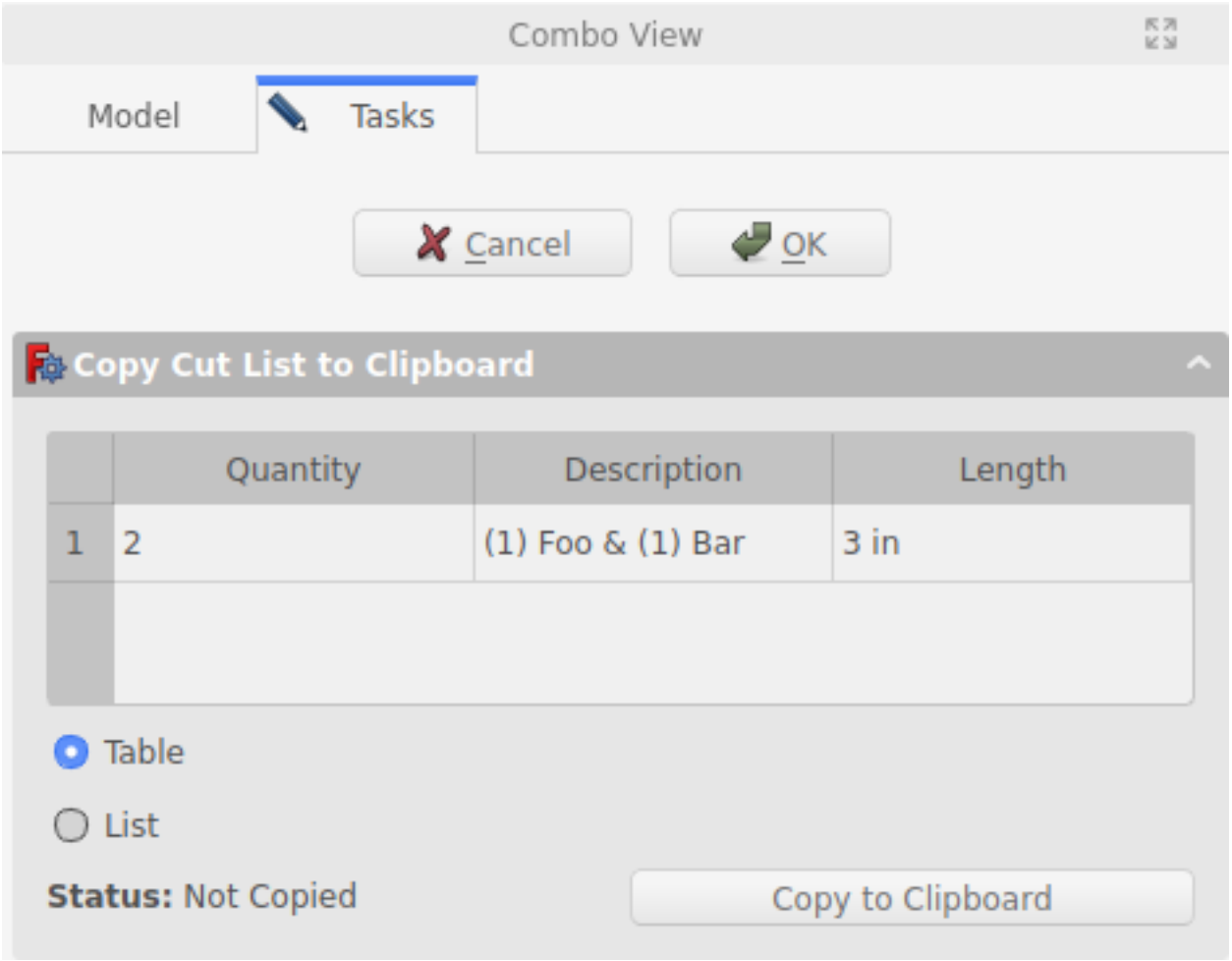

Fig. 2: Copy Cut List to Clipboard Task Panel with Merged Items

- **note** (*str, optional*) – A note to display underneath cut-list table, defaults to None

Returns Copy Cut List to Clipboard Task Panel

Return type CopyCutListToClipboardTaskPanel

create_save_cut_list_to_file_task_panel(*cut_list_items: List[osecore.gui.cut_list._create_cut_list_task_panel.cut_lis merge_cut_list_items_by_length: bool = False, note: Optional[str] = None*) → osecore.gui.cut_list._create_cut_list_task_panel.task_panel.save_cut_list_as_csv

Create a task panel to save a cut-list to the user's filesystem.

Simple Usage

```
cut_list = [
    {'quantity': '1', 'description': 'Foo', 'length': '3 in'},
    {'quantity': '1', 'description': 'Bar', 'length': '3 in'}]
panel = create_save_cut_list_to_file_task_panel(cut_list)
Gui.Control.showDialog(panel)
```

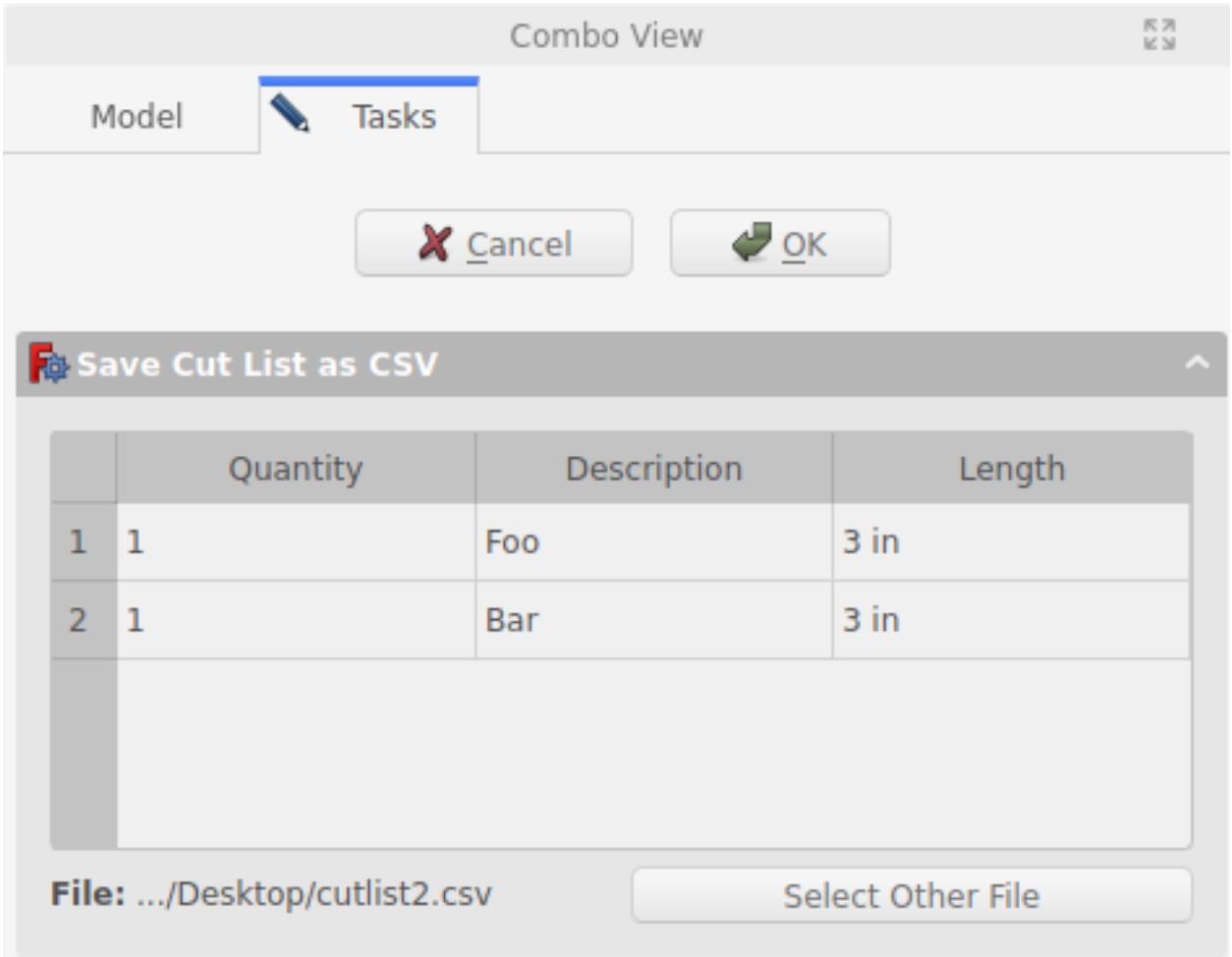

Fig. 3: Save Cut List to File Task Panel

Merge Cut List Items by Length & Note

```
cut_list = [
    {'quantity': '1', 'description': 'Foo', 'length': '3 in'},
    {'quantity': '1', 'description': 'Bar', 'length': '3 in'}]
panel = create_save_cut_list_to_file_task_panel(cut_list,
    merge_cut_list_items_by_length=True, note='example note')
Gui.Control.showDialog(panel)
```

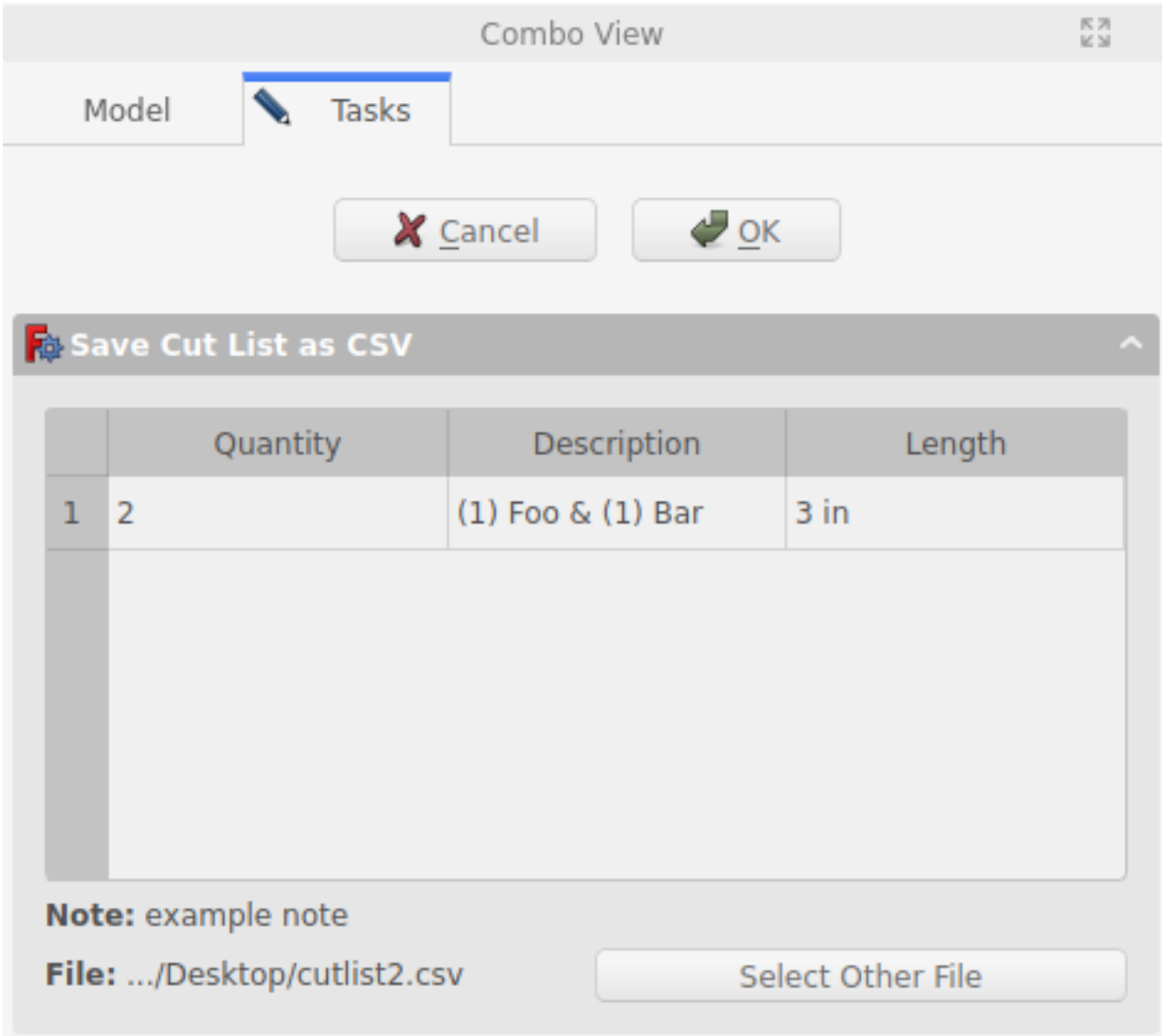

Fig. 4: Save Cut List to File Task Panel with Merged Items & Note

Parameters

- **cut_list_items** (*List[CutListItem]*) – A cut list.
- **merge_cut_list_items_by_length** (*bool, optional*) – Whether to merge cut-list items by length, defaults to `False`
- **note** (*str, optional*) – A note to display underneath cut-list table, defaults to None

Returns Save Cut List to File Task Panel

Return type SaveCutListAsCsvTaskPanel

1.2.2 osecore.gui.selection

Package for code relating to selection in GUI.

Name	Description
`find_edge_in_selection_object`	Find the first edge in the given selection object.
`find_face_in_selection_object`	Find the first face in the given selection object.
`find_selection_object_by_type`	Find the first selection object with the given object type.
`find_vertex_in_selection_object`	Find the first vertex in the given selection object.

---

find_edge_in_selection_object(*selection_object: FreeCADGui.SelectionObject*) → Optional[Part.Edge]

Find the first edge in the given selection object.

Parameters **selection_object** (`Gui.SelectionObject`) – A given selection object.

Returns The first edge found in the selection object.

Return type Optional[Part.Edge]

find_face_in_selection_object(*selection_object: FreeCADGui.SelectionObject*) → Optional[Part.Face]

Find the first face in the given selection object.

Parameters **selection_object** (`Gui.SelectionObject`) – A given selection object.

Returns The first face found in the selection object.

Return type Optional[Part.Face]

find_selection_object_by_type(*selection_objects: List[Gui.SelectionObject], object_type: str*) → Optional[FreeCADGui.SelectionObject]

Find the first selection object with the given object type.

Parameters

- **selection_objects** (`List['Gui.SelectionObject']`) – A list of selection objects returned from `Gui.Selection.getSelectionEx()`.
- **object_type** (`str`) – Type as defined on a model class bound to `obj.Proxy.Type`.

Returns SelectionObject or `None` if object with type is not found in selection.

Return type Optional[Gui.SelectionObject]

find_vertex_in_selection_object(*selection_object: FreeCADGui.SelectionObject*) → Optional[Part.Vertex]

Find the first vertex in the given selection object.

Parameters **selection_object** (`Gui.SelectionObject`) – A given selection object.

Returns The first vertex found in the selection object.

Return type Optional[Part.Vertex]

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

O

INDEX

P

Q

R

S

T

X

Y

Z