User Guide

Basic Operations
Command Reference

Reference Manual



Intelligent CAD System

ENGLISH EDITION | VERSION 1.1.4

Introduction

Thank you for choosing JDesignCAD, a general-purpose 2D and 3D CAD system.

More than 30 Years of CAD Development

Our founder has worked in the CAD industry for over 30 years,
since the early days of personal-computer CAD in the early 1990s. This experience includes quantity surveying systems for architectural CAD,
the development and support of AutoCAD drafting systems, and in-house CAD development for an electrical equipment manufacturer.
Our work has evolved alongside CAD technology throughout this period.
After establishing the company, we continued developing specialized CAD and related systems for many industries,
building a reusable library of core components from that experience.
The range of CAD systems developed in Japan has narrowed across mechanical engineering, architecture, manufacturing and civil engineering.
Although several 2D systems remain widely used,
only a small number of Japanese-developed systems also cover 3D design.

The JDesignCAD Concept

Against this background and the rising cost of purchasing and maintaining CAD software,
we developed JDesignCAD as an accessible, capable tool for small and medium-sized design firms and individual designers.
Built on our own components, it provides an integrated environment for 2D and 3D design.
Three editions are available to suit different needs: Free, STD and Pro.

Free Edition

Includes 2D drafting, creation and editing of 2D dynamic blocks, operation of existing 3D dynamic blocks, JDC saving and STL export.
Major CAD formats can also be imported. Export to DWG, DXF and other formats, and analysis functions, require STD or higher.

Standard Edition

Adds 2D and 3D stress and thermal analysis, together with export for other CAD systems
in formats including DXF, DWG, JWW, STEP, OBJ, SVG, SXF and IFC.
You can create and edit 3D face actions, run analyses and export IFC components. See the relevant chapters for the supported scope.

Pro Edition

This manual covers Free, STD and Pro. Check each chapter and the license information for feature availability.

Data Compatibility

All editions import major formats, including DXF, DWG, JWW and STEP,
and read and write JDesignCAD's native format.

Future Development

Revenue from Pro is reinvested in core performance and new functions as we continue improving JDesignCAD for design and manufacturing.

Using This Manual and General Contents

This integrated command reference covers JDesignCAD 1.1.4: basic operation, geometry editing, sheet metal, dynamic blocks, analysis, IFC/Revit, customization development and MCP integration.

In Word, follow a contents or index link to jump to a topic, normally by holding Ctrl while clicking. In a printed copy, use the page number on the right.

General Contents

Introduction 1

Part 1 Interface and Basic Settings 5
The interface, selection and input, coordinate systems, snapping, layers, models and layouts, display language and units.

Animation views 28

Add layout views 32

Orthographic views from a 3D model 33

Part 2 Ribbon Command Reference 47
Choose a command in the index to find its purpose, inputs, procedure, notes and examples. Commands follow the ribbon tabs and groups; layout commands appear at the end.

File File Operations 53

Home Draw 60
Draw clothoids 116

Basic clothoids and cross sections 117

Home Modeling 127

Home Modify 160

Home Modify Model 186

Home Annotation 220

Insert Insert 244

Insert Block Definition 248

Insert Export 251

Insert Reference 254

Manage/Measure Manage 255

Manage/Measure Measure 257

Layout Context Menu Commands 264

Part 3 Dynamic Blocks 272

Part 4 Analysis 307

Part 5 IFC Components and Revit Families 348

Part 6 Installation and Updates 366

Ribbons and display modes 41

Part 7 Sheet metal 369

Part 8 Customization guide 452

Part 9 MCP integration 459

Part 10 API reference 471

93 operations and 152 data records 509510

Register keyboard and mouse shortcuts 43

Print preview on large paper 44

Zoom Fit and complex model display 45

Calculate the volume of a solid 46

Fillet multiple edges and inspect connections 202

Move faces while retaining protrusions and holes 209

Project curves onto a face 215

Inspect slicing and interference results 199

Screenshots and Example Values

Screenshots illustrate operation. Existing figures are retained where the workflow is unchanged. Colors, labels and recent-file lists may differ in your environment.

Finding Detailed Procedures for Existing Features

This manual organizes current operation by subject. New users can start with Part 1; use the contents to find a particular task.

Language, Model Units, Fixed Units and Component Units 36
3D Dynamic Blocks, Formulas and Visibility States 273
Choosing an Analysis, Individual Materials, Beam and Thermal Exercises 308
IFC Components and Revit 2026 and 2027 Families 348
Workflow reference guide 367
Distribution Files and Editable Originals 368

Conventions

Square brackets identify on-screen commands and controls. Numerical examples explain operation and input checking; they do not specify design conditions or allowable limits for real products.

Edition Availability

Function	Free	STD／Pro
Create and edit 2D dynamic blocks	Available	Available
Operate existing 3D dynamic blocks	Available	Available
Create and edit 3D face actions	Not available	Available
Analysis tab functions	Not available	Available
Export IFC library	Not available	Available
Import SXF	Available	Available
Save SXF and export DWG, DXF and other formats	Not available	Available

Display language and model length units are independent settings. For analysis, holes and IFC, use the units shown beside each input field.

Part 1

Interface and Basic Settings

Part 1 introduces drafting in JDesignCAD: the interface, selecting geometry, changing settings, and working with models and layouts. Learn the screen and input operations first, then configure the settings needed for your work.

For detailed inputs for individual drawing and editing commands, see Part 2, Ribbon Command Reference. Part 1 covers operations and settings shared by multiple commands.

This manual covers JDesignCAD 1.1.4. The interface overview and ribbons show the current program. Existing command illustrations are retained where the workflow is unchanged. See topics 25 to 28 for language and unit settings.

Contents

01 Open a Drawing from the Start Screen 6
02 The Working Interface 7
03 Mouse and Keyboard Basics 8
04 Select Geometry 9
05 Run Commands and Enter Values 10
06 File and View Tools 11
07 Coordinate Systems and Workplanes 13
08 Snapping and Tracking 14
09 Layers and Drawing Attributes 15
10 Context Menus 17
11 The Four Properties Explorer Tabs 18
12 Project Settings 19
13 Configure Layers and Styles 21
14 Global Settings for Saving Input and Selection 22
15 Global Settings for Display and Construction Geometry 23
16 Global Settings for Snapping Grids and Approximation 24
17 Global Settings for File Exchange and Printing 25
18 Models and Layouts 26
19 Model View Settings 27
20 Layout Paper and View Scale 29
21 Layout Display Locking and Edges 30
22 Place Views on a Layout 31
23 Add Dimensions and Centerlines and Print 34
24 Initial Setup and Troubleshooting 35

01 Open a Drawing from the Start Screen

Use the start screen to create a drawing, open an existing file or browse recent files. The File tab also provides file operations while you are drawing.

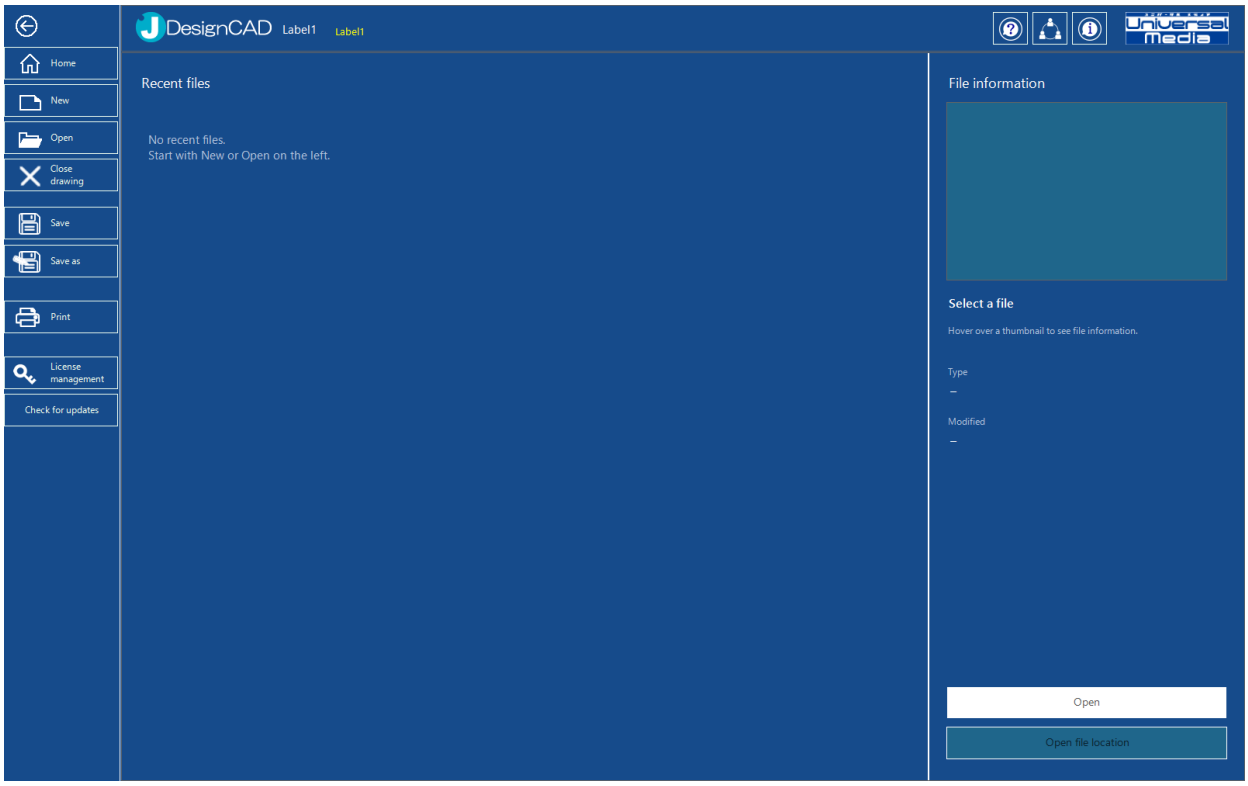


Figure 1. Start screen: file operations on the left, recent files in the center, and information about the selected file on the right.

Create a Drawing

1. Choose [New] on the left.
2. When the drawing interface opens, check the model view and workplane.
3. For the first save, choose [Save as] and specify a folder and file name. Use JDC for continued editing.

Open a Saved Drawing

Choose [Open] and select a file. For a recent file, select it, review its preview and information on the right, and click [Open]. [Open file location] opens its containing folder.

Save and Close

[Save] writes to the current file. To retain the previous version, use [Save as] with another name. When [Close drawing] prompts you to save, decide whether to keep the changes. [Print] opens output settings; check paper and scale in the preview.

The recent-file list is not the drawing data itself. If a file has been moved or deleted, open it from its actual location.

02 The Working Interface

The interface has areas for choosing commands, manipulating geometry and editing values and settings. If you are unsure what to do next, check the input fields on the right and the message at the bottom.

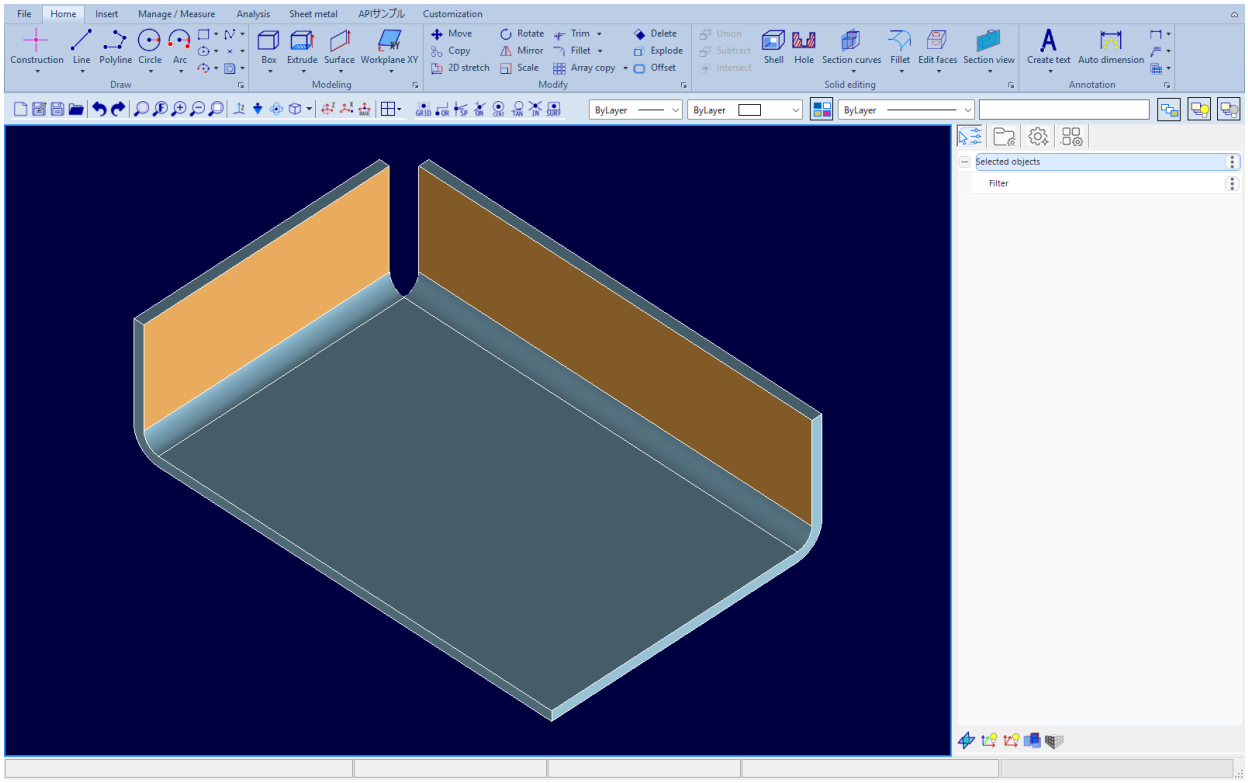


Figure 2. Working interface. The table below identifies each area.

Interface area	Location and purpose
Ribbon tabs	At the top. Switch between Home, Insert, Manage/Measure and other tabs to choose commands. File provides saving and printing.
Icon toolbar	Below the ribbon. Provides frequent file, view and snapping operations, plus layer, color and other drawing attributes.
Drawing editor	The main central area. Create and select geometry, specify positions and inspect previews in a model or layout view.
Properties Explorer	On the right. Contains command inputs, selected-object attributes, project settings, global settings and view settings.
Status bar	At the bottom. Shows the next required operation, coordinates and processing progress.

When a Command Is Unavailable

A command may be disabled because nothing is selected, the selection is not a solid, or the command cannot run in a layout. Check the current view and selection as well as the command name.

Selection, pending-input and workplane colors depend on settings and operation state. Check object names and input fields as well as colors.

03 Mouse and Keyboard Basics

Distinguish editing geometry from moving the view to avoid unintended changes. To pan, drag with the wheel button instead of left-dragging geometry or handles.

Operation	Main function	Notes
Left-click	Select geometry, specify a point or position, or select a command or input field.	The object requested depends on the active command.
Left-drag	Select a region or manipulate selected geometry and handles.	Check whether the drag begins on geometry or empty space.
Rotate the wheel	Zoom around the pointer.	This does not change the model's physical size or the layout's print scale.
Drag with the wheel button pressed	Pan the view.	Use this to inspect another area while zoomed in.
Ctrl and wheel-button drag	Rotate the model view.	This is separate from changing the workplane.
Right-click	Open a menu for the current object or operation.	Menus differ for geometry, empty space, view frames and property entries.

Confirm and Cancel

During drawing, use the check mark at the top of Properties Explorer to confirm or the cross to cancel. The status bar indicates when Enter or Esc is available. In continuous drawing or view insertion, objects already created may remain after you end the command.

While editing a numerical field, Enter confirms that value. Check the preview and status message to determine whether the entire command has finished.

Repeat a Command

After a command ends or is canceled, press Enter or Space when the drawing editor has focus to restart the last repeatable command. It begins with fresh input rather than reprocessing the previous selection.

One-time actions such as Delete, Cut and file operations are not repeatable this way. View-specific commands refer to their original view; select the command again from another view when working there.

Undo an Operation

Use [Undo] on the toolbar to reverse the last edit, and [Redo] to restore it. To cancel incomplete input, first cancel the active command. Undo is not a substitute for saved backups.

04 Select Geometry

Normally, click an object to select it. Use Ctrl or drag a rectangle from empty space to select multiple objects. While a command is active, only the object types required by that command can be selected.

Selection method	Operation	Objects selected
Single selection	Left-click an object.	Selects the clicked object, normally replacing the previous selection.
Add or remove	Hold Ctrl and click an object.	Adds it to the selection. Clicking an already selected object removes it.
Window selection	Drag from left to right, starting in empty space.	Selects objects entirely inside the rectangle.
Crossing selection	Drag from right to left, starting in empty space.	Selects objects inside or crossing the rectangle.
Add a region	Hold Ctrl while selecting a region.	Adds objects while retaining the existing selection.
Cycle overlapping objects	Hold Alt and click near the same location.	Cycles through overlapping candidates. Check the highlight before confirming.

Before Selecting

1. Cancel any unwanted active command to return to normal selection.
2. Check the object's layer and whether that layer is locked.
3. If necessary, check selection modes in the context menu and property filters.
4. After selecting, check the highlight and the object type shown in Properties Explorer.

Solids Faces and Edges

A whole solid, a face and an edge are different selection targets. Moving a solid and filleting an edge require different selections. A visible line may be a separate 2D entity, a 3D edge or a projected layout outline; the available commands differ.

When Selection Fails

Zoom in, then check selection mode, filters and layers. In a layout, determine whether you are selecting a view frame or using a command that targets edges inside it. A visible outline is not always an independently editable line.

05 Run Commands and Enter Values

Most commands start from the ribbon and accept successive inputs in the panel on the right. The preview lets you check those inputs; seeing a preview does not necessarily mean that creation or editing has been committed.

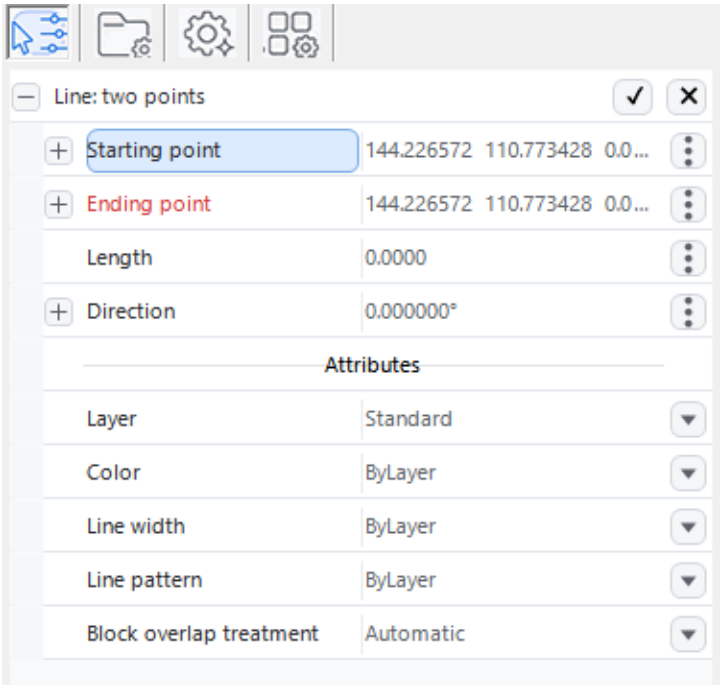


Figure 3. Input panel for a line start and end. The check mark and cross at the upper right confirm or cancel.

Basic Workflow

1. Choose a ribbon command. Expand buttons with a small arrow to choose related construction methods.
2. Check the panel on the right for the point, object or value currently requested.
3. Specify points and objects in the drawing editor. For an exact value, select the value field, type the number and confirm it.
4. Check position, orientation, size and operation mode in the preview.
5. Complete the required inputs, then use the check mark or the command's confirmation action.

Panel entry	Meaning
Name on the left and value on the right	The left identifies the required input; the right shows its value. Expand coordinates and directions to see their components.
Plus and minus signs	Expand or collapse child inputs. Collapsing an entry retains its values.
Arrow and vertical-dot buttons	Open choices, additional menus or object-selection options. Contents vary by entry.
Red input entries	Indicate an input being requested. Also check the status bar for the required action.

Some fields, such as radius and diameter, specify the same dimension in different ways. Choose the appropriate field and check both the value and the preview.

06 File and View Tools

The icon toolbar below the ribbon provides frequent operations. Hover over an icon to see its name. Button arrangements may vary with window width and configuration.

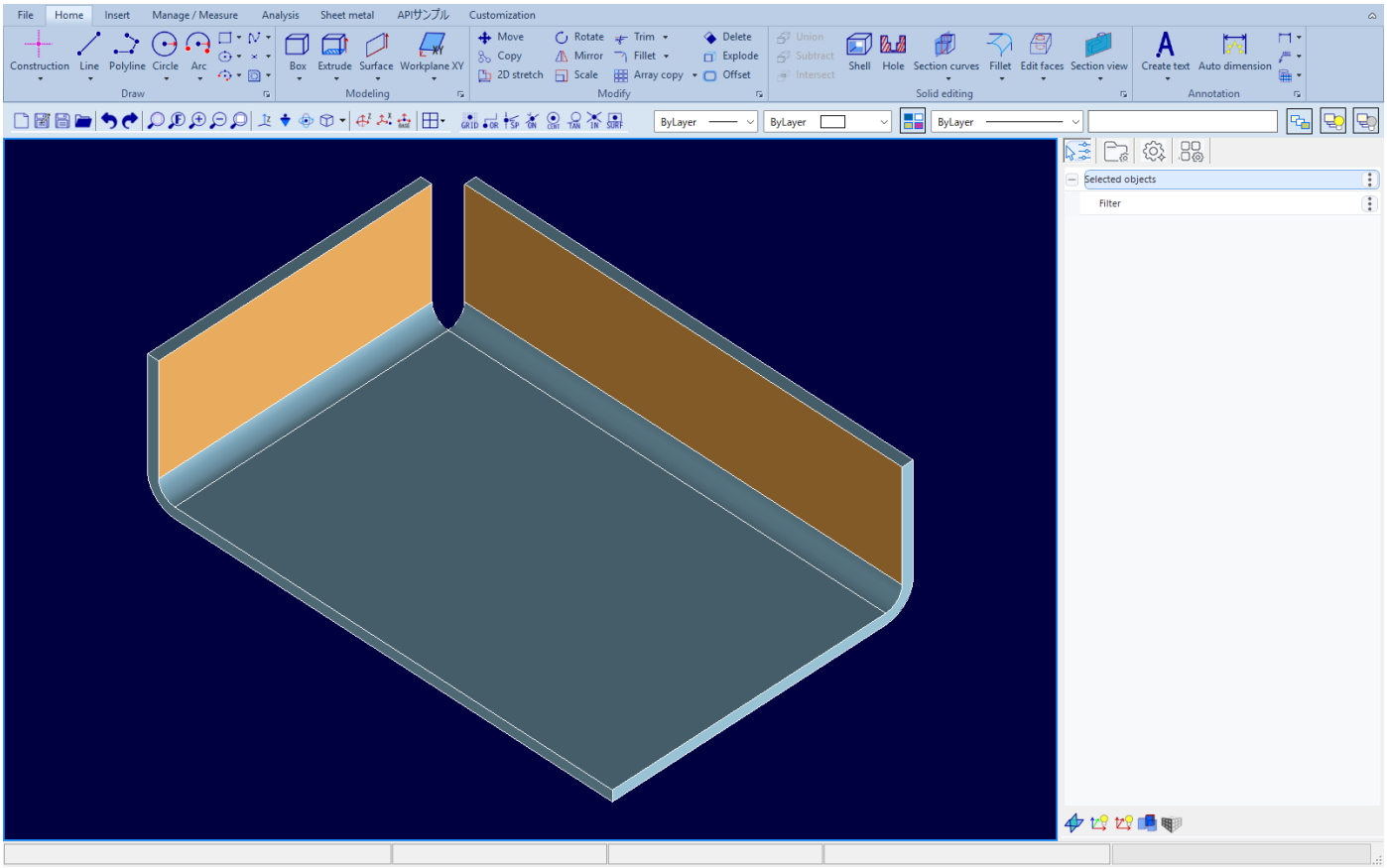


Figure 4. Left side of the toolbar: file and view tools, followed by workplane and snapping controls.

Command	Purpose
New / Open	Create a drawing or open a saved drawing.
Save as / Save	Save under another name or save the current file. Use another name to retain the earlier state.
Undo / Redo	Reverse an edit or restore an undone edit.
Redraw	Refresh the display, for example to clear display remnants.
Zoom extents	Adjust the view to show all geometry.
Zoom in / Zoom out	Change the display magnification in steps.
Zoom window	Zoom to a rectangle specified by two opposite corners.
Z axis up	Adjust the view orientation so the Z axis points upward.
View rotation center	Set the pivot for view rotation. Useful when inspecting a small feature.
Rotate view	Change the viewpoint with the mouse. This does not rotate the geometry itself.
View direction	Choose top, front, right, left, isometric or workplane views.
Split model view	Inspect the same model in one to four viewports. This is separate from creating three orthographic layout views.

When Geometry Disappears After Zooming

Try [Zoom extents]. If the geometry still appears too small, check for distant objects and visible layers. You do not need to scale the actual geometry to find it.

Display-only Operations

Zooming, panning and changing viewing direction do not alter model geometry or dimensions. Changing a layout view's scale or projection direction does alter its appearance on the sheet. Check which setting you are editing.

07 Coordinate Systems and Workplanes

WCS is the drawing's global coordinate system; UCS is the system for the current work. Moving the workplane to a side or inclined face changes its X and Y directions and the perpendicular Z direction.

Set the Drawing Plane First

To draw 2D lines or circles on the side of a 3D model, set the workplane first. Choosing a front or side view does not necessarily change the workplane. Display the workplane and UCS symbol to check their orientation and origin.

Operation or display	Purpose
Choose a workplane	Use ribbon workplane commands to choose a standard plane or a plane based on geometry. See the workplane commands in Part 2 for details.
Reverse UCS Z axis	Reverse the front/back orientation of the workplane, for example when checking an extrusion direction.
UCS X direction	Set the X direction within the workplane.
Set UCS XY origin	Set the origin used on the workplane.
Workplane view direction	View the current workplane from a convenient direction.
WCS / UCS in the status bar	Check the same position in global and current working coordinates.

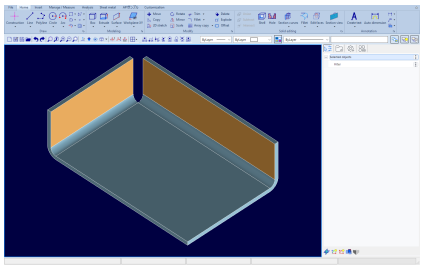


Figure 5. Display toggles at the bottom of Properties Explorer.

Toggle Reference Displays

The lower buttons show or hide the workplane, WCS symbol and UCS symbol. Making non-active layers transparent helps identify the working layer. Hiding a display does not delete coordinates or geometry.

Switching between parallel and perspective projection changes the view. Use orthographic views for reading dimensions and perspective views for presenting the shape.

Tracking Directions

Tracking uses directions in and perpendicular to the current workplane. The blue Z tracking line indicates UCS Z, which may differ from WCS Z on a side workplane.

Create a UCS from the WCS

Z Flip, UCS Origin, and X Direction on the coordinate-system toolbar can be used even when the XY WCS is active. For X Direction, specify a point indicating the direction from the current origin. Enable polar tracking to track along the configured directions. Canceling retains the original coordinate system.

08 Snapping and Tracking

Snapping helps specify exact endpoints, centers and other geometric points. Check that the desired snap is active rather than merely clicking nearby.

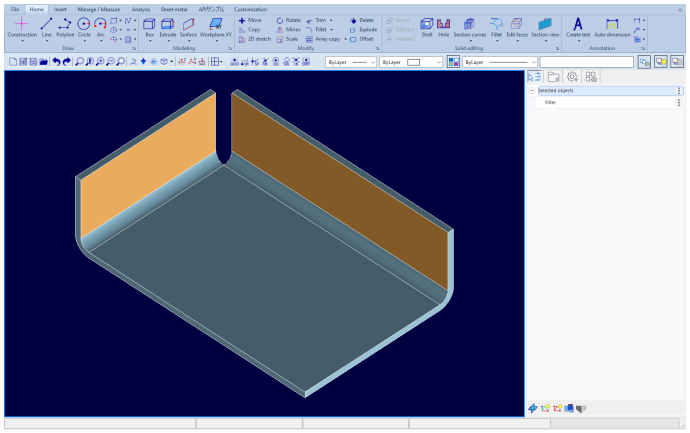


Figure 6. Snap toggles. Enable the types required for the task.

Button name	Typical use
Grid	Snap to configured grid points. This is independent of grid visibility.
Ortho	Draw in orthogonal directions from a reference point.
Endpoint / Object snap point	Specify characteristic points of an object, such as line endpoints.
Point on object	Specify a point along a line or curve.
Center	Specify the center of a circle or arc, rather than a point on its circumference.
Tangent	Use a tangent location. Available candidates depend on the command and geometry.
Intersection	Specify where objects intersect.
Point on face	Specify a position on a 3D face.

Enable Only the Snaps You Need

Too many candidates can cause snapping to a nearby unintended point. Favor Center for center-to-center dimensions or Endpoint for drawing from an endpoint, and temporarily disable unnecessary types.

Use Tracking

Under Global settings > Snap settings, configure polar tracking, polar angle, object snap tracking and Z tracking. Use the guides to align a direction or position from a known point.

The polar tracking angle determines candidate directions; it does not rotate existing geometry. If guides point in unexpected directions, check the workplane and enabled snaps.

When a Visible Grid Does Not Snap

Grid visibility and grid snapping are independent. Enable the toolbar's grid snap as well as the grid display. Check horizontal and vertical spacing under grid settings.

09 Layers and Drawing Attributes

Color, line pattern and line width can be assigned individually or inherited from a layer. For consistent drawings, create layers by purpose and set object attributes to ByLayer.

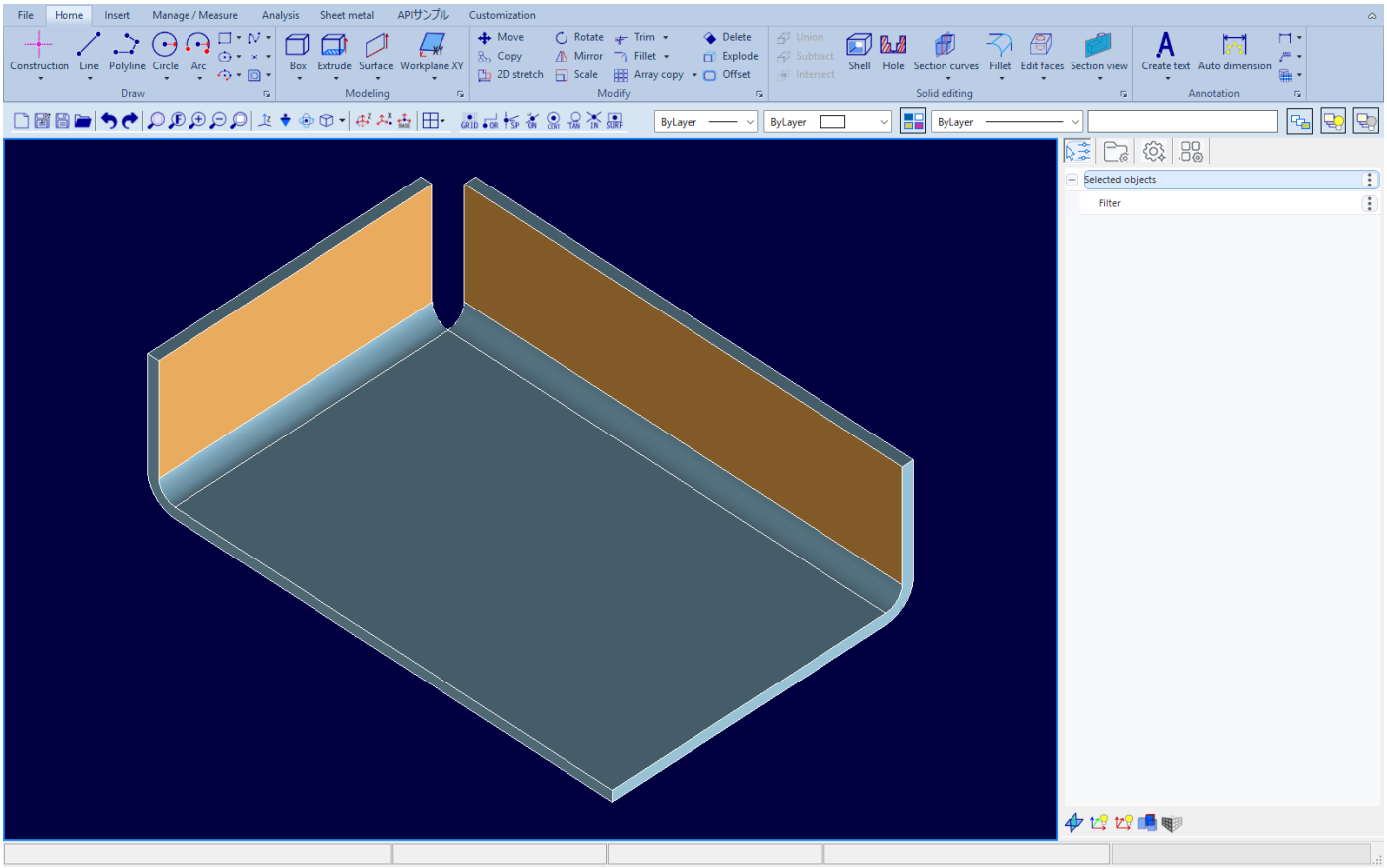


Figure 7. Attribute controls on the right of the toolbar: layer, color, line width and line pattern.

Attribute	Purpose
Layer	Organize geometry by purpose, such as outlines, dimensions, centerlines and construction geometry.
Color	Set the display color. Check print settings to determine whether it prints in that color.
Line pattern	Set a solid, dashed, dash-dot or other pattern.
Line width	Set line thickness. Its appearance depends on display and print settings.
ByLayer	Inherit color, line pattern and line width from the object's layer.

New Geometry and Selected Geometry

With nothing selected, attribute controls set the properties of new geometry. With objects selected, the same controls may change their attributes. Clear the selection before changing only the defaults for new objects.

Draw on Layers Organized by Purpose

1. Create the required layers in Project settings > Layer table.
2. Set each layer's color, line pattern and line width.
3. Make the required layer current and set color, pattern and width to ByLayer.
4. Draw a test line and check its layer and attributes.
5. To standardize existing objects, select them and set their layer and ByLayer attributes.

Changing a layer's color does not affect objects with individually assigned colors. The same applies to line patterns and widths. If a layer setting is not reflected, check the object attributes.

Layers added or renamed in Project Settings appear in the toolbar list. Layer Manager provides List and Image views, visibility, locking, current layer, color, line pattern and line width. Visibility changes apply to the active view.

Centerline Attributes

New centerlines are placed on the Centerline layer with color, line pattern and line width set to ByLayer. If absent, the layer is created with the dimension color and CENTER pattern. An existing Centerline layer keeps its settings.

10 Context Menus

Context menus depend on where you click and what is selected. Use them for commands directly related to geometry, views or individual properties.

Right-click location	Main operations	What to check
Empty drawing area	Selection modes, Paste and display operations.	The menu differs between an active command and normal selection.
Selected geometry	Cut, Copy, Delete, Move, Rotate, Scale and Explode.	Available commands depend on the selected object type.
Solid face or edge	Edit or create using faces and edges.	Check whether the target is the entire solid, a face or an edge.
Layout view	Add projected views, control edges and centerlines, show or lock views.	Select the intended view before running the command.
Property entry	Add or delete entries, choose options, or run setting-specific commands.	Check whether you are acting on a parent list or an individual entry.

Basic Operation

1. Select the target in normal selection or within the intended command.
2. Right-click it and choose a menu command.
3. Follow the property inputs and status-bar guidance when the command starts.
4. Dismiss unwanted menus. If an entry is disabled, check the target and selection conditions.

Common Layout Operations

[Hide edges] removes unwanted edges from that view's drawing representation; it does not delete 3D edges. [Change selected edge width] adjusts line thickness on the sheet without changing model attributes.

To hide a whole view, toggle [Show view]. Restoring a hidden view is separate from restoring hidden edges within it.

When the Menu Is Unexpected

Clear and reselect the target, or cancel the active command before right-clicking. In a layout, distinguish lines drawn directly on the sheet from model outlines inside projected views.

11 The Four Properties Explorer Tabs

Use the four icons at the top of Properties Explorer to switch its contents. Distinguish settings for the current drawing from application-wide settings.

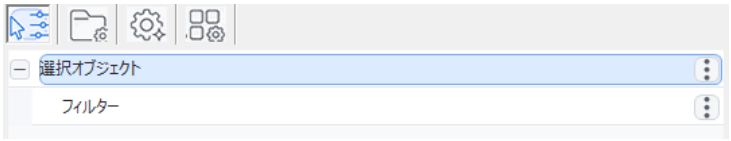


Figure 8. Tabs from left to right: Selection and Operations, Project Settings, Global Settings, and Views.

Tab	Contents	Applies to
Leftmost Selection and Operations tab	Selected-object information and active-command inputs.	Selected geometry or the operation being entered.
Second Project Settings tab	Colors, layers, line patterns and dimension styles.	The currently open drawing.
Third Global Settings tab	Interaction, display colors, snapping, autosave and output settings.	Application behavior. Attribute lists also provide defaults for new drawings.
Rightmost Views tab	Model views and layouts, viewpoints, paper and view scale.	The specified view or layout.

Read the Leftmost Tab

With nothing selected, this tab shows selected-object lists and filters. Selecting geometry shows its type and attributes. Starting a command replaces this area with that command's input panel.

For example, selecting a line shows endpoints and attributes; drawing a circle requests its center and radius. Read the panel title to distinguish editing an existing object from defining a new one.

Edit Settings

Expand entries with the plus sign and edit the value to the right of the name. Arrow fields offer choices; vertical dots and right-click menus provide more actions. Confirm numerical input with Enter or the relevant control.

If an entry is missing, check the tab, expand its parent and scroll. Global Settings contains additional entries farther down the list.

Show the Input Panel Automatically

Check Global settings > Operation > Automatically show operation panel. If inputs are not visible when you start drawing, switch to the leftmost tab.

12 Project Settings

Project Settings manages attributes and styles registered in the current drawing. These are saved with the drawing. When opening another drawing, check the settings belonging to that file.

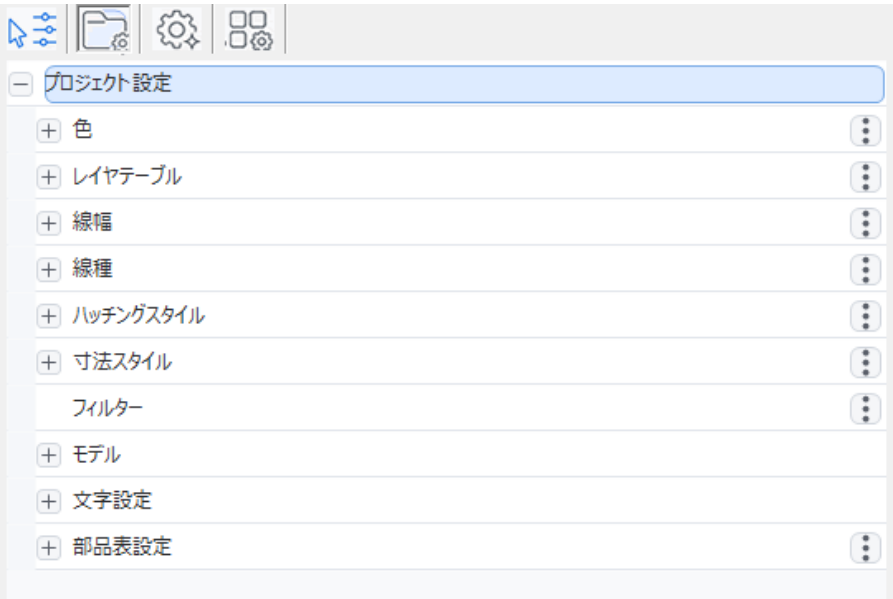


Figure 9. Project Settings. Expand an entry to inspect its definitions.

Item	Settings
Color	Colors used in the drawing. Inspect existing colors or register new ones.
Layer table	Layers and their attributes for organizing geometry and controlling visibility and editing.
Line width	Line widths available in the drawing. Register the required thicknesses.
Line pattern	Definitions such as Continuous, CENTER and HIDDEN, applied to objects or layers.
Hatch styles	Manage fill and hatch patterns and attributes.
Dimension styles	Manage dimension text, arrows, lines and display conventions together.
Filters	Manage selection conditions based on object types and attributes.
Models	Manage drawing models. This differs from Views, which manages viewpoints.
Text settings	Set text size, font, bold, italic and other formatting.
Parts list settings	Configure columns and other parts-list settings.

Current Drawing Versus New Drawings

Use Project Settings to change this drawing's line patterns and dimension styles. To set initial attribute lists for new drawings, use the corresponding Global Settings lists. Changing those lists does not replace settings in every existing drawing.

Match component colors and linetypes to the destination

Settings inside the block	Attributes after insertion
Layer 0 + ByLayer	Color, linetype, and lineweight of the insertion layer
Layer 0 + ByBlock	Attributes of the placed block reference
A layer other than 0 + ByLayer	Attributes of the original layer

Settings inside the block	Attributes after insertion
Explicit color, linetype, and lineweight	Retain the specified values

To make a general-purpose component adopt the destination layer's color, create its contents on layer 0 with ByLayer attributes. Layer 0 is reserved and cannot be renamed, deleted, or removed by PURGE. Objects on layer 0 outside a block use the attributes of layer 0 itself.

13 Configure Layers and Styles

Before drawing, establish a small set of layers and representative styles to simplify later consistency changes. When editing an existing drawing, first check its conventions.

Prepare Layers

1. Expand [Layer table] in Project Settings.
2. Add a layer from the list menu and give it a descriptive name.
3. Set its color, line pattern and line width. If drawing is unavailable, also check locking and visibility.
4. Choose the target layer on the toolbar and draw with ByLayer attributes.

Example layer name	Example settings
Outlines	Use continuous lines with a consistent width for principal outlines.
Dimensions	Use colors and widths distinguishable from outlines. Also check the dimension style.
Centerlines	Use CENTER. Retain the settings of any existing centerline layer.
Construction geometry	Keep it separate so it is easy to hide when finishing the drawing.

These are examples. Choose names and attributes to match your drafting standards. Hiding a layer is different from deleting geometry.

Check Line Patterns and Widths

Check that the required pattern exists under [Line patterns], then assign it to a layer or object. Verify its name in Properties Explorer as well as its appearance. For widths, check both the model view's line-width mode and print settings.

To add a line width, create an item under Project Settings > Line Widths, then set both its name and numeric width in millimetres. For example, name it 0.5mm and set Width (mm) to 0.5. Naming an item 0.5mm does not assign that physical width.

The attribute list refreshes after additions, renaming and value changes. A custom zero-width item shows its name and 0mm. ByLayer uses the layer setting; ByBlock uses the inserted block setting. Line pattern name and pattern edits also refresh the list.

Standardize Dimensions and Text

Expand [Dimension styles] and check the text, arrows and lines of the chosen style. Create a representative dimension to verify readability on the sheet before applying it throughout.

For ordinary text, check size and formatting under [Text settings]. Text and dimensions have separate settings, so changing ordinary text size does not necessarily change dimension text.

Change an Existing Drawing

Changing a style or layer can affect many objects that reference it. Save a separate copy first, then inspect the model, layouts and print preview after making changes.

14 Global Settings for Saving Input and Selection

Global Settings controls common JDesignCAD behavior. Its upper entries define initial colors, layers, widths, patterns, hatches, dimensions, text and parts lists for new drawings. Interaction and display settings follow below.

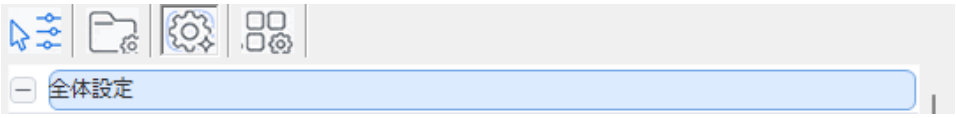


Figure 10. Global Settings. Attribute lists, autosave, display, snapping and printing continue down the panel.

Autosave

Use [Enable autosave] to turn it on or off and set the interval in minutes and destination. Choose a writable folder with sufficient space that you can easily find later.

Autosave writes timestamped JDC backups of changed drawings; it does not overwrite the original file. It may wait during drawing or processing, and may skip empty or unchanged drawings. Continue saving manually at suitable intervals.

Display Format

Set the decimal separator and decimal places for general values, coordinates, vectors and angles. Radius/Diameter controls circle input presentation. Reducing displayed decimals does not round the model geometry. Also check dimension styles for dimension formatting.

Selection and Operation

Setting group	Adjustments
Selection size and sensitivity	Handle size, drag threshold and selection-box display. Check these when small objects are difficult to edit.
Selection colors and appearance	Selection boxes, handles, selection and focus colors, highlight width, opacity and display position.
Preselection highlight	Candidate highlighting and its color. Make it distinguishable from selected objects.
Automatically show operation panel	Control input-panel display when starting a command.
Delete 2D sketch after solid creation	Controls whether the source 2D sketch is retained. Check this if you want to reuse the source profile.

15 Global Settings for Display and Construction Geometry

Display settings improve working visibility. Choose background and reference colors that keep geometry, selection highlights and tracking lines distinct.

Coordinate Symbols and Colors

Under [Display], check WCS and UCS symbol visibility and position. [Color settings] controls background, grid, drawing aids, layout background, workplane color and transparency, and the active-view frame.

Item	When to adjust
Background / Layout background	Improve visibility in model space and on the sheet. Change object colors through layer or object attributes.
Grid / Drawing-aid colors	Distinguish grids and input guides from actual geometry.
Workplane color / Transparency	Make the plane's location clear while keeping the model behind it visible.
Active-view frame color	Distinguish the active viewport or view frame from its background.

Construction Lines

Set the length factor, width and color of construction lines used for alignment and drafting references. See Part 2 for creating, deleting and showing these lines.

Construction lines and centerlines have different purposes. Create permanent drawing centerlines with the centerline commands and their dedicated layer and line pattern.

Other Reference Settings

Settings group	Purpose and use
Interference check	Set interference-solid and contact-face colors and result layers so results are distinguishable from the original model.
Block attributes	Control whether the attribute-input dialog appears when inserting an attributed block.
Balloons and markers	Set initial symbol, prefix, suffix, numbering method and default attributes for part numbers and similar annotations.
Font display	Adjust display mode and accuracy. This is separate from the font and size assigned to drawing text.

16 Global Settings for Snapping Grids and Approximation

Point-input settings and curve-approximation settings serve different purposes. Do not try to solve selection or direction problems by changing approximation accuracy or maximum path gap.

Snap Settings

Item	Function
30-degree / 45-degree snap	Enable input aids for the corresponding angular directions.
Snap to workplane origin	Include the current workplane origin as a point candidate.
Snap to world origin	Include the WCS origin as a point candidate. It may differ from the UCS origin.
Polar tracking	Enable directional alignment from a reference.
Polar tracking angle	Set the angular interval for direction candidates.
Object snap tracking	Use object snap points to guide alignment.
Z tracking	Guide alignment perpendicular to the workplane.

Endpoint, Center, Intersection and other snap types can also be toggled on the toolbar. Enable the required types first, then use tracking for additional directional guidance.

Grid

Set grid spacing with [Horizontal distance] and [Vertical distance], and choose its appearance under [Display]. Model views also have grid settings; check the target model view under Views when adjusting the current screen.

Grid spacing is a drafting aid. Choosing a spacing of 10 does not change all dimensions to multiples of 10. Enable Grid snap to snap to the grid.

Approximation and Paths

Setting	Meaning and notes
Approximation accuracy	Set accuracy for curve approximation. This does not control decimal display or globally change the accuracy of all solid operations.
Curve approximation method	Choose the representation used for approximation. Inspect the result before accepting a conversion of the original curve.
Maximum path gap	Set the gap criterion for joining elements into a path. Avoid unnecessarily large values that could join unrelated objects.

Leave approximation and gap settings unchanged for ordinary drafting unless a particular operation requires adjustment. Retain a separate copy of the drawing before testing changes.

17 Global Settings for File Exchange and Printing

Configure file exchange for the receiving application and drawing purpose. Keep an editable JDC original and export exchange files separately so the source information remains available.

Export Settings and STEP Import

Item	What to check
DXF / DWG format and export version	Choose a format and version supported by the receiving software.
DXF / DWG substitute fonts and text	Check font substitution and text handling. Inspect Japanese text and symbols after export.
Dimension import / export	Configure dimension handling. Check appearance and editing in the receiving application too.
Multithreaded STEP import	Control parallel processing during import.
Merge similar STEP faces	Control face consolidation during import. Face and edge structures may change.
Create STEP blocks	Organize imported data into blocks.
Prefer non-periodic STEP surfaces	Control surface representation during import. Keep the original file when comparing settings for problematic data.

Print

Item	Purpose
Print method	Choose a rendering/output method. The applicable subordinate settings depend on this choice.
Color output	Check color or other output modes. This is separate from the screen background color.
Default line output width	Set the printed width for lines using the default width.
OpenGL line-pattern pitch factor	Adjust pattern repetition spacing for OpenGL output.
GDI line-pattern pitch factor	Adjust pattern repetition spacing for GDI output.
GDI shading	Control shaded output with GDI.
GDI bitmap printing	Control rasterized printing with GDI.
GDI overlap handling / Draw order	Adjust overlapping geometry output and drawing order.

18 Models and Layouts

A model contains geometry drawn and edited at full size. A model view is a working view of that model. A layout places model views on paper at suitable directions and scales to form a finished drawing.

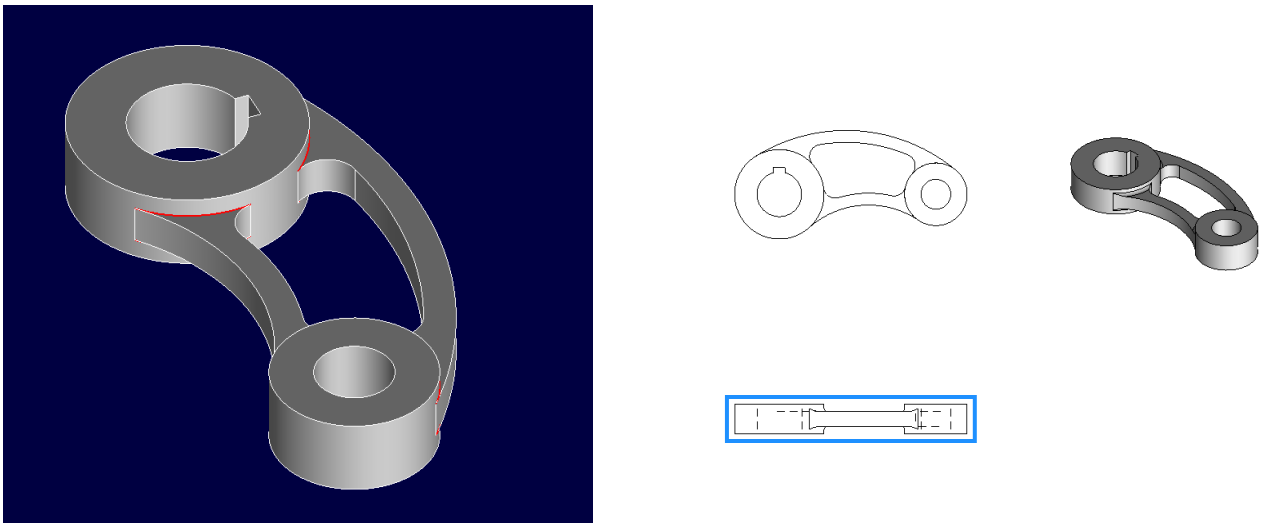


Figure 11. Model view on the left and layout on the right. Red edges on the left indicate selection.

Comparison	Model and model view	Layout
Main purpose	Create full-size geometry and edit its shape and position.	Arrange paper, projections, scales and annotations for output.
Zooming	Changes the visible area, not the actual dimensions.	Changes the sheet's on-screen size; view scale is a separate setting.
Showing the same geometry	Inspect one model from several directions.	Place front, top, detail and other views of the same model.

Projected views reference the source model, so model changes appear in related views. Dimensions and centerlines need associative references to follow changes. Lines and text drawn directly on the sheet are separate layout entities.

Create Several Views of One Model

Place front, top and pictorial views of the same model. You do not need to copy the component model to add views to the sheet. Each view can have its own display mode and scale.

Geometry Drawn Directly on the Sheet

Place titles, notes and supplementary lines on the layout sheet. They are separate from model geometry. When moving a projected view, check whether sheet annotations are associated and will follow it.

19 Model View Settings

Expand a model view under Views to set its viewpoint, workplane and display. Project Settings manages models; View settings manages how they are seen.



Figure 12. Expanded model-view settings.

Item	Purpose
Projection direction / Distance	Set the direction and relative viewpoint position. First check a standard view command before editing numerical direction values.
Workplane	Check or set the current drawing plane. This is separate from viewing direction.
Line-width mode	Control how line widths are displayed, including whether they are used.
Light intensity	Adjust shaded-view brightness. This does not change material properties or model geometry.
Light position	Choose a lighting direction that makes protrusions, recesses and holes easy to inspect.
Grid	Check grid settings for this model view.
Visible layers	Choose the layers displayed in this view.
Display mode	Choose wireframe, shading or another model display mode.

Use Split Views

Choose horizontal, vertical or standard four-view splits from the model-view split control. The views share one model, so no geometry duplication is needed. Click the required viewport and check its active frame before working.

Each viewport has its own zoom and viewpoint. Splitting arranges the working screen; it does not create sheet projections. Switching to a layout returns to a single display. Create permanent drawing views in the layout.

Inspect motion in an animation view

An animation view displays model objects moving along specified axes or curves. It does not calculate dynamics from forces and masses, and it is separate from deformation playback in the analysis window.

Prepare fixed and moving objects, then create an animation view from the view list. Check layer visibility and add a Curve Drive or Axis Drive. Assign a path to a curve drive, or an axis location and direction to an axis drive.

Assign only moving parts under Moved Objects, leaving fixed parts unassigned. The path or axis and the moving geometry are separate selections. In this example only a slider moves along a straight path; set its start position to zero and check direction.

Create a schedule and add drive, time and position entries. For example: position 0 at 0 seconds, 40 at 2 seconds and 0 at 4 seconds. Activate the intended schedule before starting playback from the view menu. Curve-drive position is distance along the path.

Choose once, infinite repeat or a repeat count. Stopping restores the original positions, and switching to another view stops playback. Save as JDC and check the active schedule and assigned objects after reopening. Playback does not automatically stop at collisions or contact.

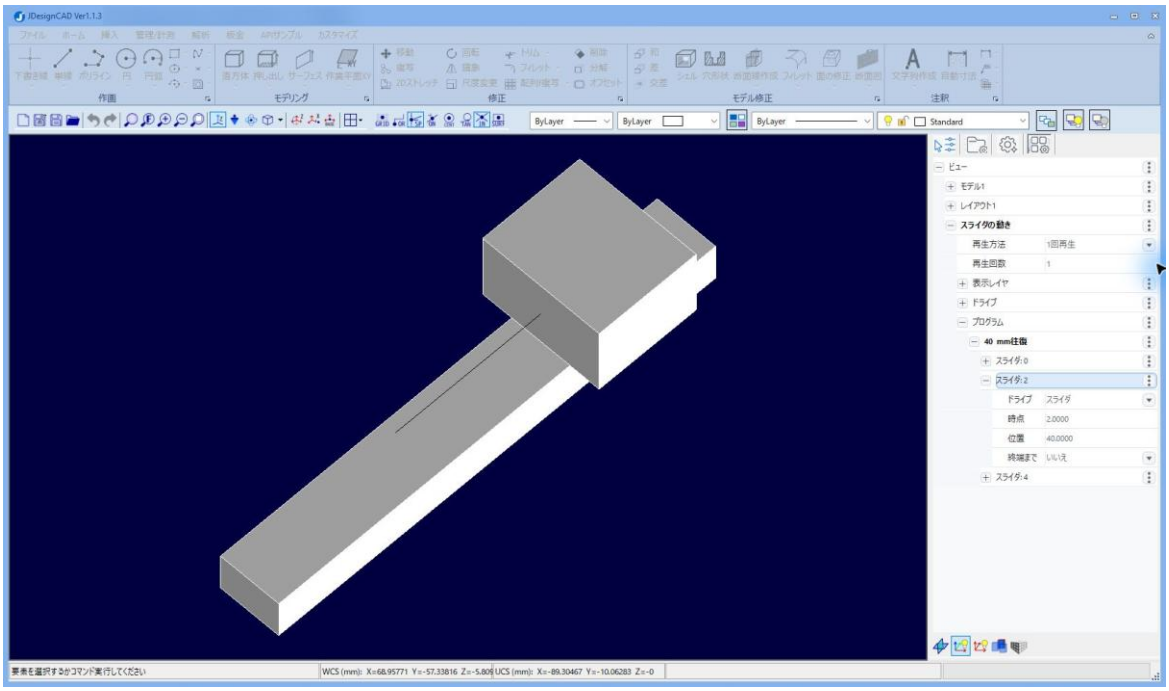


Figure 203 Linear drive and slider example

20 Layout Paper and View Scale

Expand a layout under Views to inspect its paper and contained views. Sheet settings and individual-view settings are at different levels.

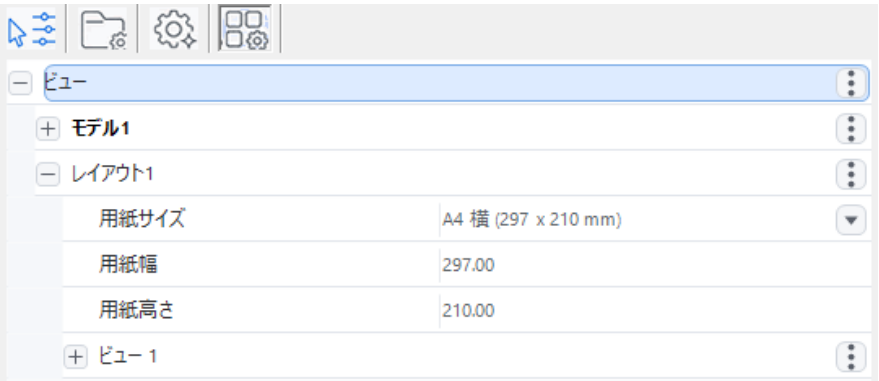


Figure 13. Layout paper settings. Expand View 1 below to inspect its individual settings.

Set the Paper

1. Display the target layout and expand it under Views.
2. Choose [Paper size] and check [Paper width] and [Paper height].
3. Arrange views with enough space for their number, scale, dimensions and title block.
4. Check that printing uses the same paper size and orientation.

Distinguish Three Types of Size

Size type	Where to change it	What changes
Actual model size	Drawing and model-editing commands.	Physical dimensions of the component or geometry.
View scale	The Scale setting of each layout view.	Magnification of the model on paper.
Screen magnification	Mouse wheel and view zoom commands.	Only the displayed size on the working screen.

A model length of 100 mm appears as 50 mm on paper at 1:2 or 200 mm at 2:1. Associative dimensions read the actual model size. Scaling again during printing applies another factor, so check the print settings.

View Position and Extents

Each view has model-side and layout-side positions, plus width and height. Distinguish the model and layout qualifiers after Left and Bottom. Before fine adjustment, decide whether you want to move the frame or its contents.

Geometry outside the view frame is not displayed. If a view appears clipped, check its extents. Changing view scale does not change the model's actual dimensions.

21 Layout Display Locking and Edges

Individual layout views can have display modes and edge widths independent of the model. Use these view settings to refine drawing appearance without changing geometry.

Item	Purpose and notes
Show view	Show or hide the view. Hidden views are also excluded from printing and 2D export.
Lock view	Protect sheet position, extents, scale and projection direction. Dimension snapping and editing sheet text remain available.
Visible layers	Set which layers appear in this view.
Projection direction and type	Set viewing direction and parallel or perspective projection.
Brightness and light position	Improve visibility in shaded views.
Display mode	Choose wireframe, shading or another display mode.
Show drawing hidden lines	Control hidden lines in drawing display.
Hidden-line pattern and color	Set the pattern and color of hidden lines.
Show frame	Toggle the frame indicating the view's extents.
Lock internal movement	Control how the frame and its contents move relative to each other. This differs from protecting placement with [Lock view].

Change Edge Widths

For the whole view, enter a paper-space width in [Edge width (mm)]. Zero uses original model attributes. This does not change model-edge attributes or other views.

For individual edges, enter [Selected edge width (mm)] and choose [Change selected edge width] from the entry or view context menu. Click edges or select a rectangle, then press Enter. Assigning zero restores the view-wide setting for those edges.

Hide Unwanted Edges

Choose [Hide edges] from the view context menu, select the edges and press Enter. Esc cancels selection. Hiding applies only to this view and does not delete model edges.

When a View Is Missing

Hidden views remain in the settings tree. Restore visibility there or use the layout's [Show all views]. Text and lines drawn directly on paper are separate and may remain visible when a view is hidden.

22 Place Views on a Layout

Layouts can contain base, projected, section and detail views. Choose the final paper size and scale first to make later arrangement easier.

Create Projections from a Base View

1. Prepare the model and switch to the target layout.
2. Run [VIEWBASE (Base view)] from the layout context menu.
3. Check orientation, scale, display mode and first-angle or third-angle projection.
4. Click on the sheet to place the base view.
5. Move above, below or to either side of the base view and click to place the required projections.
6. Press Enter or Esc to finish adding views. Esc before placing the base view cancels creation; ending afterward retains views already placed.

Add Views Later

Select an existing view and choose [Add projected view] from its context menu. New views inherit its model, scale and projection method. Check that the source view references the intended model.

View type	Purpose
Base view	Place the reference orientation, such as the front view.
Projected view	Add top and side views relative to the base to describe the shape from several directions.
Auxiliary view	Show an inclined face or another direction not clear in standard front and top views.
Section view	Show a cut representation to explain internal geometry.
Detail view	Enlarge a selected area to explain small features.

Follow the input panel and status bar for each view command. Changing a section or detail view is separate from cutting or scaling the original 3D model.

Check Placement

Use descriptive view names and check sheet boundaries and spacing between views and dimensions. Once position and scale are set, lock the view to prevent accidental changes while annotating.

Add another view to a layout

Open the layout and add a view from the view list or layout menu. Set its model, orientation, scale, paper rectangle and visible layers. Give it a name, hide unwanted layers and lock it before adding dimensions and notes.

An ordinary added view has independent orientation and scale. To arrange aligned orthographic views from a front view, use base and projected views below. Placing a view on paper does not create another 3D part in model space.

Create three orthographic views from a 3D model

The example is an 80 × 50 × 20 mm part with a hole. Check the solid, switch to a landscape A4 layout and start Base View (VIEWBASE) from the layout context menu.

Choose the model, front direction, 1:1 scale, hidden-line display and third-angle projection. Place the front view toward the lower left, then move upward for the top view and rightward for the right-side view. First-angle projection places these corresponding views on the opposite sides; set the required projection convention first.

Finish additional placement with Enter or Esc. To add views later, start Projected View (VIEWPROJ) from an existing base or projected view. It inherits the source scale and projection convention; an independent view or perspective view is not a valid parent.

Before dimensioning, check orientation, hidden hole edges, alignment and scale in all three views. Dimension actual projected geometry and size text for printing. Use text leaders for hole counts, radii and chamfers. After model changes, review views and dimensions, then check line weights, text and paper margins in Print Preview.

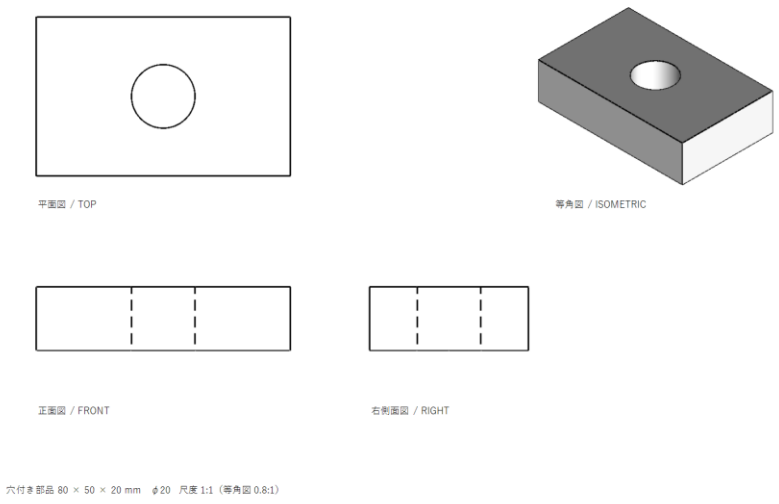


Figure 201 Orthographic and isometric views of a part with a hole

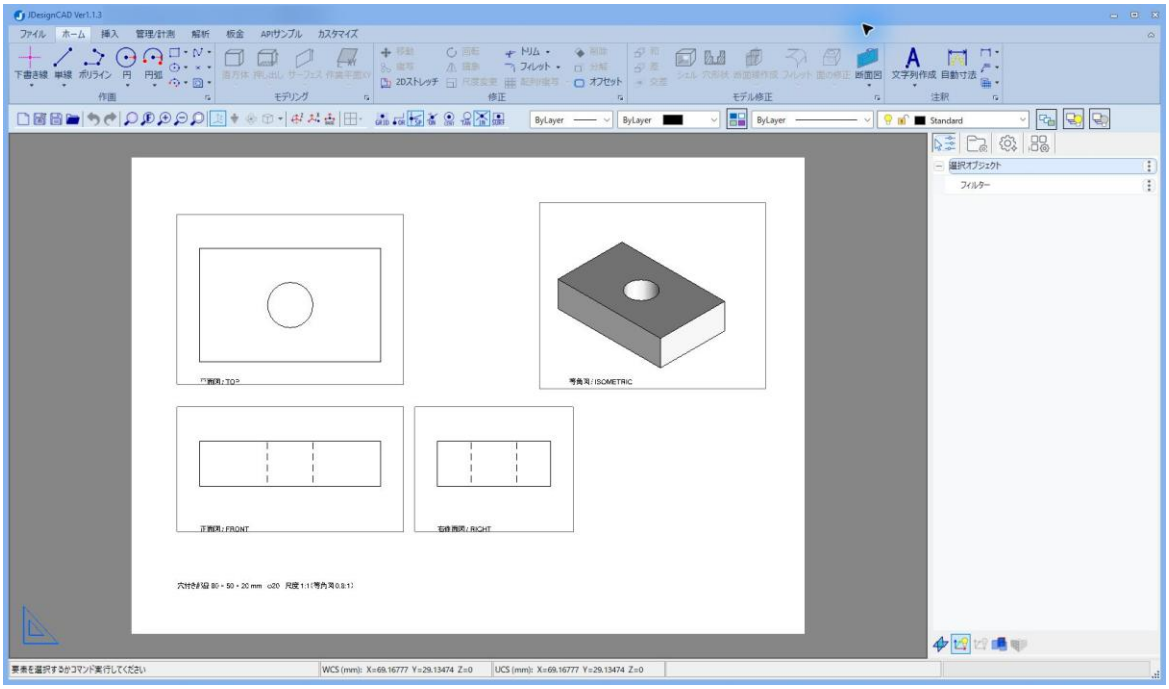


Figure 202 A layout containing several views

23 Add Dimensions and Centerlines and Print

Layout dimensions must reference the correct circles, endpoints or other projected geometry. Picking only an apparent paper position may give incorrect physical dimensions or prevent updates from following the model.

Add Dimensions

1. Set view orientation and scale, and lock the view if needed.
2. Start the dimension command and select geometry or snap points within the view.
3. For center-to-center dimensions, use Center snap. For diameter or radius, select the relevant circle or arc.
4. Place the dimension and compare it with a known model dimension. Check references and scale before overwriting the displayed number.

Add Centerlines

Open the Line drop-down on Home and choose [Circle centerline] or [Centerline between two lines]. Select a circle or arc, or two parallel lines. In a layout, select objects within the same view.

Target	Association behavior
Projected layout view	Creates a centerline referencing the source circle or lines and the view. You can snap to it for dimensioning.
3D edges in a model view	Creates a centerline at the 3D location, referencing the edges.
Independent 2D circles or lines	Creates a non-associative centerline that does not automatically follow source changes.

[Extension length (paper mm)] is a paper-space length independent of view scale. Centerlines are created on the Centerline layer with ByLayer attributes. The original edges need no color or width changes.

Even associative references can fail if their source is deleted or extensively modified. After model edits, check centerline positions and dimension values.

Before Printing

1. Check paper size and orientation, view scales, and geometry or dimensions extending beyond the sheet.
2. Check the global print method, color output, line widths and line patterns.
3. Use File > Print to inspect the preview, including hidden views and edges.
4. For accurate physical output, ensure the printer does not apply additional fit-to-page scaling. If necessary, measure a known dimension on a test print.
5. Save the JDC and export a separate exchange drawing if required.

24 Initial Setup and Troubleshooting

Start with settings directly relevant to drafting. Adjust detailed approximation, import and print-method settings once basic work is established.

Initial Setup Sequence

1. Choose a folder and drawing name, then save a JDC first.
2. Check autosave in Global Settings and set its folder and interval.
3. Check the model view, workplane, UCS origin and directions.
4. Prepare layers and line patterns; verify ByLayer color, pattern and width.
5. Enable required snaps such as Endpoint and Center. Configure grid display separately from snapping.
6. Check the readability of the background, selection colors, text and dimensions.
7. For printing, set paper and view scale and preview representative lines, text and dimensions.

Troubleshooting by Symptom

Symptom	First checks
Cannot select geometry	Check the active command, selection mode, filters, layer visibility and locking, and the target within the view.
A command does not proceed	Check required inputs, confirm numerical entries and read the status bar. A preview alone may not complete the operation.
Settings do not affect geometry	Check Project versus Global Settings, ByLayer versus fixed object attributes, and the target view's settings.
Grid is visible but does not snap	Check Grid snap on the toolbar; it is separate from visibility.
Point directions or heights are wrong on a side face	Check the workplane and UCS origin and directions, not just the viewing direction.
Cannot move a layout view	Check view locking and whether you are manipulating the frame or its contents.
Dimension values differ from actual size	Check reference objects, Center snapping, view scale and dimension associations.
Printed lines are thick or dashes are crowded	Check object, layer and view widths, then the print method, default output width and line-pattern pitch.
Cannot find the drawing	Check Zoom extents, visible layers, hidden views and the actual saved file location.

Verify the effect of each setting change before proceeding. See Part 2 for command input sequences and detailed options.

Language and Units Contents

Configure display language and drawing units.

25 Change the Display Language 37
26 Model Input and Display Units 38
27 Inputs with Fixed Units 39
28 Component and Exchange-file Units 40

25 Change the Display Language

JDesignCAD supports Japanese and English for the ribbon, start panel, main dialogs and tooltips. This edition uses English operation names; retained screenshots may show Japanese labels.

Procedure

1. Save your current drawings.

2. In Properties Explorer > Global Settings, select Japanese or English as the language.

3. Close and restart JDesignCAD. The new display language takes effect after restarting.

4. Check the ribbon and start panel, then open your drawing and continue.

Language and Units Are Independent

Selecting English does not automatically change the model to inches. Set model units under Project settings > Models. You can also work in inches with the Japanese interface.

Item	Effect of changing language
Ribbon and main settings dialogs	Displayed in the selected language.
Drawing component and layer names	User-assigned names are retained.
Drawing text and additional product specifications	Entered text is retained and is not automatically translated.
Coordinates lengths angles and geometry	Not changed by display language alone.
Analysis and IFC fields with explicit units	Use the units displayed in the field.

Reading the Screenshots

Basic-operation and command illustrations include earlier versions with unchanged workflows. Colors, button positions, labels and recent-file lists may differ from your environment. The text defines current operation and supported scope.

26 Model Input and Display Units

Model units specify how lengths are entered and displayed. Changing them does not scale existing geometry. Ordinary model coordinates are stored internally in millimeters.

Procedure and Example

1. Open Project settings > Models > the target model, for example Standard model.

2. Under [Units], choose millimeters, centimeters, meters, kilometers, inches or feet.

3. Check length and coordinate displays. A line physically 25.4 mm long reads 1 in inches or 0.0254 in meters.

4. Ordinary length entries are converted from the chosen model unit to physical length. Entering 2 in an ordinary length field with inch units gives 50.8 mm.

Quantity	Conversion rule	Example for a 25.4 mm cube
Lengths and coordinates	Convert by the length factor.	Side length 1 inch.
Area	Use the square of the length factor.	645.16 mm^2 = 1 $inch^2$
Volume	Use the cube of the length factor.	16,387.064 mm^3 = 1 $inch^3$
Angles counts ratios and percentages	No length conversion.	90 degrees, 3 items and a factor of 2 remain unchanged.

Changing the Actual Drawing Size

To change a physical length from 100 mm to 200 mm, use Scale or a suitable dimension-driven edit. Switching from mm to cm only displays 100 mm as 10 cm. Older JDC drawings also retain their original physical size.

Difference from Paper Scale

Model units, actual model size and layout scale are independent. Displaying the model in inches does not change paper size or a print scale such as 1:10. Dimensions inside models and blocks inherit model units.

27 Inputs with Fixed Units

Check each field's label and unit before entering a value. Even with an inch-based model display, fields explicitly labeled mm, N or another unit require that unit.

Setting or operation	Input and display in version 1.1.4
Ordinary model coordinates lengths and dimensions	Model input/display units.
Area and volume	Model units squared or cubed.
Paper size and positions on paper	mm
Dimension-style text and arrow sizes	Millimeters; distinguish these from actual model dimensions.
Global text height and default grid spacing	Fields labeled mm use millimeters.
Line width line pattern and display settings	The mm or px unit shown in each field.
Hole diameter and depth	Use the indicated millimeters.
Analysis geometry dimensions and displacement	Millimeters. Loads and material properties use their labeled units.
IFC port position diameter width and height	Millimeters. A direction vector is not a length.

Common Analysis Unit Confusions

Force uses N; total load uses N or kgf; surface load uses N/mm²; line load uses N/mm. Young's modulus and stress in N/mm² are numerically equivalent to MPa. Temperature uses °C, heat flux W/m², and heat-transfer coefficient W/(m²K).

JDesignCAD does not switch analysis inputs to inch, lbf or psi. Even for models displayed in inches, enter the mm and SI values specified in the analysis dialog. Imported geometry uses its physical dimensions.

Checking Sequence

After changing units, measure a known length, check dimension labels and inspect the layout print preview. Do not additionally scale geometry by 25.4 merely because the displayed unit changed.

28 Component and Exchange-file Units

JDC components used by Block Manager are based on millimeters. They retain their physical size even when inserted into a model displayed in inches. Distinguish an exchange file's unit metadata from the drawing's display units.

Format or object	Import or insertion	Export
JDC components	Retain physical dimensions based on mm.	Save JDC as the editable original.
DWG／DXF	Convert using file units; if unspecified, use the global import-unit setting.	Coordinates and header in mm; block-definition base units retained separately.
External blocks with units	Retain definition base units and convert at insertion.	Retain the relationship between definition units and insertion scale.
STEP／SAT	Convert from file unit information.	Check the chosen export format's settings and receiving application.
STL／OBJ	Set [File length unit] in Global Settings.	Export in millimeters.

Before Opening STL or OBJ

These files normally contain no length-unit information. Set Global settings > STL / OBJ import > File length unit to match the source before opening. The default is mm; choose meters if the source uses meters.

When Physical Size Is Incorrect

1. Check a known dimension and the unit selected when exporting from the source application.

2. Measure the same feature after import. A factor such as 25.4 or 1,000 suggests a unit mismatch.

3. Reopen a unitless file with the correct import unit. Changing only model display units cannot repair incorrectly imported physical dimensions.

Dynamic-block definition units and lookup formulas are retained. Resetting an externally modified block for editing restores its initial appearance, so save the original as JDC first.

Ribbon and display modes

Use the current ribbon images below to locate commands. The API sample tab appears after installing the supplied sample. The customization management group is named Extensions.

Tab	Purpose
Home	Draw, model, modify, direct editing and annotation
Insert	Blocks, dynamic definitions and export
Manage / Measure	Drawing/display management and measurement
Analysis	2D/3D analysis. STD/Pro
Sheet metal	Rules, creation, editing, unfolding/refolding and flat patterns
API sample / Customization	Sample commands, add-in status and installation folder

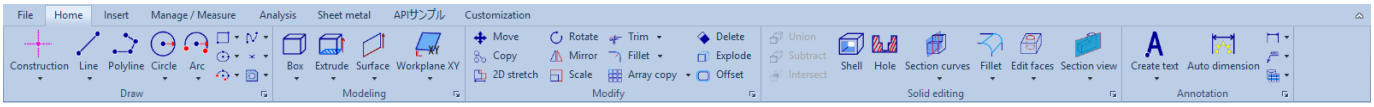


Figure 1 Actual current ribbon. Groups may collapse differently with window width.

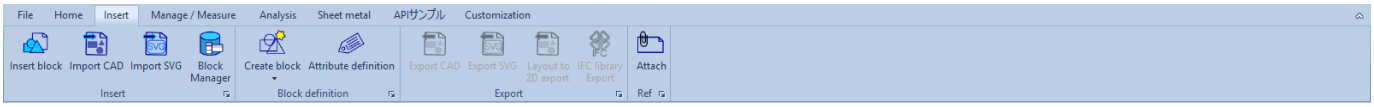


Figure 2 Actual current ribbon. Groups may collapse differently with window width.



Figure 3 Actual current ribbon. Groups may collapse differently with window width.



Figure 4 Actual current ribbon. Groups may collapse differently with window width.

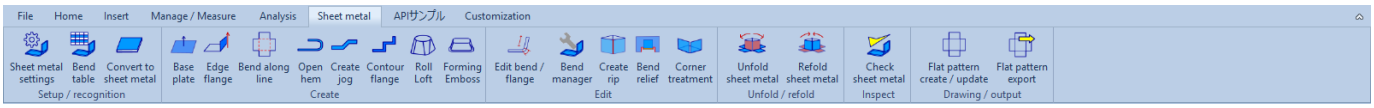


Figure 5 Actual current ribbon. Groups may collapse differently with window width.



Figure 6 Actual current ribbon. Groups may collapse differently with window width.

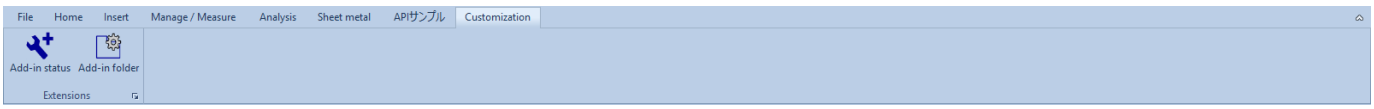


Figure 7 Actual current ribbon. Groups may collapse differently with window width.

Text visibility and shaded occlusion

Filled text is drawn without lighting, retaining its specified color in isometric views. Solid faces retain lighting. Existing shaded modes remain, with a new shaded/no-edges hidden-line mode.

Mode	Appearance
Shading	Shaded faces with model edges
Shading without edges	Hide model edges while keeping 2D entities in front
Shading without edges with hidden lines	Hide model edges and occlude 2D entities behind faces

Use the added occlusion mode for lines representing model details so hidden lines do not show through faces. Use the existing mode when annotations should stay in front.

Shortcuts and printing checks

Register keyboard and mouse shortcuts

Open Manage / Measure, then Shortcut registration in the Manage group. Filter the command list by name or command ID and select a command.

Choose a key and Ctrl, Shift or Alt modifiers. Use Press a key to capture a keyboard key or mouse side button, then choose Register / Change. Confirm replacement if the key already has an assignment.

Choose OK to save and apply. Cancel or the window close button discards changes made in the dialog. Removal and reset also require OK. Import and export use JSON.

Supported inputs include F1 through F24, letters, numbers, the numeric keypad, arrows, Delete and XButton1/XButton2. Reserved keys such as Esc, Enter, Space, Tab and Alt+F4 cannot be assigned. Text, numeric and table input and other dialogs take priority over command shortcuts.

Shortcuts are per-user settings, not drawing data. Assignments survive a Japanese/English language change. Other running CAD processes load the settings when next started.

Print preview on large paper

Select paper size, orientation and scale, then update the preview. For A1, A0 and other large sizes, verify the physical printer supports the selected size and orientation.

If the driver substitutes another size, the preview uses the requested paper and reports the mismatch. Printing is unavailable in that state. Select a supported printer or paper size and update the preview again.

At a fixed scale, geometry may extend beyond the sheet. Choose Fit to paper or an appropriate scale to include the drawing. Preview support does not expand the physical printer paper range.

Zoom Fit and complex model display

Show the layers you want to see and run Zoom Fit. Objects on hidden layers, including 2D geometry, are excluded from the displayed bounds. If geometry appears missing, check layer visibility, selection filters and the active model.

Complex 3D drawings require display geometry calculation on the first opening, after an engine update, or after drawing content changes. Reopening an unchanged drawing can use the display cache. Preparation time depends on the geometry and computer.

For Move or Copy, specify the base point and destination, review the preview and confirm. Input previews update through a display transform; actual geometry is processed on confirmation. Check the resulting position and number of copies, using Undo if necessary.

Drawing loading uses saved display information and valid display caches. Switching between model and layout views reuses valid display data. Changed drawings and missing caches still require display preparation.

Zoom Fit uses visible geometry. Hide unwanted 2D layers before fitting a model. MCP uses view.activate to change views and document.open to open an existing native drawing.

Calculate the volume of a solid

Select a solid and double-click the Volume label in its properties, or choose Calculate volume from that row menu. Wait for the calculating indication to become a value.

Volume calculation starts explicitly so that selecting a complex shape does not repeatedly trigger expensive work. Not calculated does not by itself indicate an invalid solid. Other operations remain available during calculation.

The result is a numerical approximation. A millimetre drawing displays cubic millimetres; volume follows the model display-unit conversion. Reselecting a solid retains the value. Recalculate after geometry edits or reopening the drawing.

Part 2 Ribbon Command Reference

Command Index

The number on the right is the command's page. If names repeat, also check the tab, group and command ID on the reference page.

File File Operations

New 53
Open 54
Save as 56
Save 57
Close drawing 58
Print 59

Home Draw

Polyline 60
Construction line Cross 61
Construction line Horizontal 62
Construction line Vertical 63
Construction line Two-point angle 64
Construction line Angle 65
Construction line Relative angle 66
Construction line Point and tangent 67
Construction line Two tangents 68
Construction line Fixed reference offset 69
Construction line Moving reference offset 70
Construction line Delete all 71
Construction line Show or hide 72
Line Single line 73
Line Continuous lines 74
Line Length and angle 75
Line Endpoint and tangent 76
Line Two tangents 77
Line Tangent and angle 78
Line Perpendicular bisector 79
Line Angle bisector 80
Line Parallel by distance 81
Line Parallel through point 82
Line Perpendicular through point 83
Line Multiline 84
Line Circle centerline 85
Line Centerline between two lines 86

Home Draw

Circle Center and point 87

Circle Center and radius ... 88
Circle Two points ... 89
Circle Three points ... 90
Circle Two points and radius ... 91
Circle Three tangents ... 92
Circle Two tangents and point ... 93
Circle Two tangents and radius ... 94
Circle Tangent point and radius ... 95
Circle Tangent center locus and radius ... 96
Arc Center and point ... 97
Arc Center and radius ... 98
Arc Two points and radius ... 99
Arc Three points ... 100
Arc Three tangents ... 101
Arc Two tangents and point ... 102
Arc Two tangents and radius ... 103
Arc Tangent point and radius ... 104
Rectangle Opposite corners ... 105
Rectangle Center width and height ... 106
Rectangle Width height and angle ... 107
Rectangle Two points and height ... 108
Parallelogram ... 109
Polygon ... 110
Ellipse Three points ... 111
Ellipse Two points and two directions ... 112
Elliptical arc Three points ... 113
Elliptical arc Two points and two directions ... 114
Spline ... 115
Helix ... 118
Point ... 119
Divide ... 120
Hatch ... 121
Simple hatch ... 122
Boundary ... 123
Check closed geometry ... 124
Image ... 125
Image by two points ... 126

Home Modeling

Solid Box ... 127
Solid Cylinder ... 128
Solid Cone ... 129
Solid Sphere ... 130

Torus 131
Extrude 132
Revolve 133
Loft 134
Path loft 135
Extrude 136
Revolve 137
Planar surface 138
Loft 139
Path loft 140
Network 141
XY workplane 142
Offset XY workplane 143
XZ workplane 144
Offset XZ workplane 145
YZ workplane 146
Offset YZ workplane 147
Offset workplane 148
Face on model 149
Face edge and angle 150
Path and curve plane 151
Workplane by three points 152
Workplane perpendicular to path 153
WCS X angle 154
WCS Y angle 155
WCS Z angle 156
UCS X angle 157
UCS Y angle 158
UCS Z angle 159

Home Modify

Move 160
Copy 161
2D stretch 162
Rotate 163
Mirror 164
Scale 165
Delete 166
Explode 167
Offset 168
Trim 169
Surface trim 170
Split 171

Join ... 172
Remove overlapping objects ... 173
Extend ... 174
Envelope ... 175
Circular clip ... 176
Rectangular clip ... 177
Fillet ... 178
Fillet all corners ... 179
Chamfer ... 180
Chamfer all corners ... 181
3D fillet ... 182
3D chamfer ... 183
Rectangular array ... 184
Circular array ... 185

Home Modify Model

Union ... 186
Subtract ... 187
Intersect ... 188
Shell ... 189
Hole Basic Shape and Entrance ... 190
Hole Repeated Clicks and Array Placement ... 191
Hole Thread Information and Simplified Display ... 192
Hole Automatic Thread-depth Calculation ... 193
Hole Double-click Editing ... 195
Section curves ... 196
Split solid by faces ... 197
Solid interference check ... 198
Fillet ... 200
Chamfer ... 203
Remove fillet ... 204
Change fillet size ... 205
Remove faces ... 206
Move faces ... 207
Rotate faces ... 210
Copy faces ... 211
Offset faces ... 212
Taper faces ... 213
Project curves onto faces ... 214
Create section plane ... 216
Toggle live section ... 217
Section curves to XY plane ... 218
Section settings ... 219

Home Annotation

Automatic dimension 220
Text 221
Text by two points 222
Convert text outlines to paths 223
Multiline text 224
Horizontal 225
Vertical 226
Length 227
Between objects 228
Radius 229
Diameter 230
Angle between objects 231
Three-point angle 232
Four-point angle 233
Chamfer dimension 234
Leader 235
Surface Label 236
Coordinate leader 237
Balloon 238
Marker 239
Geometric tolerance 240
Datum 241
Surface finish symbol 242
Parts list 243

Insert Insert

Insert block 244
Insert CAD geometry 245
Insert SVG geometry 246
Block 247

Insert Block Definition

Attribute definition 248
Create block 249
Export block 250

Insert Export

Export CAD geometry 251
Export SVG geometry 252
Export layout as 2D 253

Insert Reference

Attach 254

Manage/Measure Manage

Purge 255
Reference manager 256

Manage/Measure Measure

Distance between two points 257
UCS angle 258
Face angle 259
Closed-geometry perimeter 260
Area and section properties 261
Distance between faces 262
Solid volume and centroid 263

Layout Context Menu Commands

Layout Display and View Management 264
Layout VIEWBASE Base View 265
Layout Add Projected View 266
Layout Auxiliary View Normal to a Face 267
Layout Add Circular Detail View 268
Layout Section View 269
Layout Hide and Restore Edges 270
Layout Change Edge Widths 271

Ribbon tab: File	Group: File Operations
 Command: New	Command ID: File.New

Create a new drawing.

Procedure

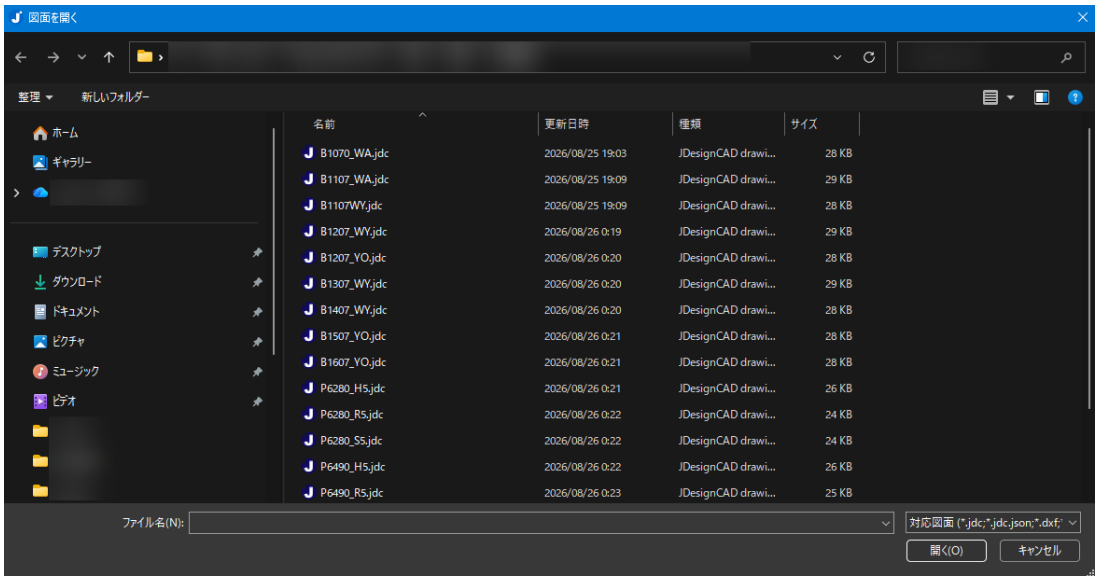
① Immediately after startup, this command switches from the File tab to Home.

② If a drawing is already open, another JDesignCAD window starts with a new drawing.

Ribbon tab: File	Group: File Operations
Command: Open	Command ID: File.Open

Open drawings or models in JDC, DXF, DWG, JWW, STEP, STL and other formats.

Procedure
The Open dialog appears.



③ Immediately after startup, the command switches from File to Home and opens the data.
④ If a drawing is already open, another JDesignCAD window starts and opens the data.

SXF P21 and SFC Exchange

SXF supports 2D exchange for civil engineering and similar drawings. Free can import it; STD and Pro can save and export it. JDesignCAD exports SXF 3.1.

Procedure

1. In Open or Insert CAD geometry, choose SXF P21 (*.p21) or SXF SFC (*.sfc).

2. Check known dimensions, layers, line patterns, text and dimension placement. Black geometry is displayed in black or white to suit the background.

3. For export, prepare 2D geometry on the active model's XY plane and choose P21 or SFC from Save as or Export.

4. Also save JDC to retain editing information. You can continue editing the original drawing after saving SXF.

Object	Exchange behavior
Standard length angle arc-length radius and diameter dimensions	Editable dimensions where measurement references and placement can be reproduced.
Special leaders tolerances and similar annotations	Exploded to lines and text where equivalent dimensions cannot be reproduced.
Hatches	Retained where supported. Unsupported patterns become lines, or outlines with a warning.
Multiline text	Exported as single-line text while preserving line placement.
Fills	Converted to planar regions, including holes.
Zero line width	Exported as 0.01 mm; source width is retained.

Supported Scope

This does not automatically project 3D or inclined geometry onto XY, nor claim complete SXF compatibility or OCF certification. Review warnings and check dimensions, text and scale in the receiving application.

Dimensions already exploded into lines and text by older versions cannot automatically regain measurement references. Re-export the editable JDC original with the current version.

Ribbon tab: File	Group: File Operations
 Command: Save as	Command ID: File.Save.As

Save the current drawing with a specified name and format.

Procedure
Save the open file under a chosen name.

The Free edition supports JDC in this save dialog.

STD and Pro can save JDC, DWG, DXF and JWW.
3DSOLID geometry is written as mesh data in DWG and DXF.
To transfer solid geometry to AutoCAD, use Export CAD geometry to write STEP,
then import it using AutoCAD's STEPIN command where supported.

JWW supports 2D geometry only.

Ribbon tab: File	Group: File Operations
 Command: Save	Command ID: File.Save

Save the drawing over its existing file.

Procedure

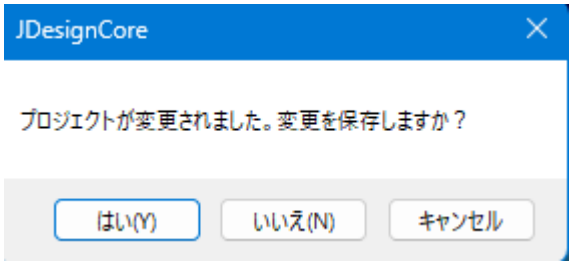
If the file has already been saved, it is overwritten.
If it has not yet been saved, Save as opens instead.

Save as also opens for drawings imported from formats other than JDC.

Ribbon tab: File	Group: File Operations
Command: Close drawing	Command ID: none (internal operation)

After checking whether to save, close the drawing and return to the start screen.

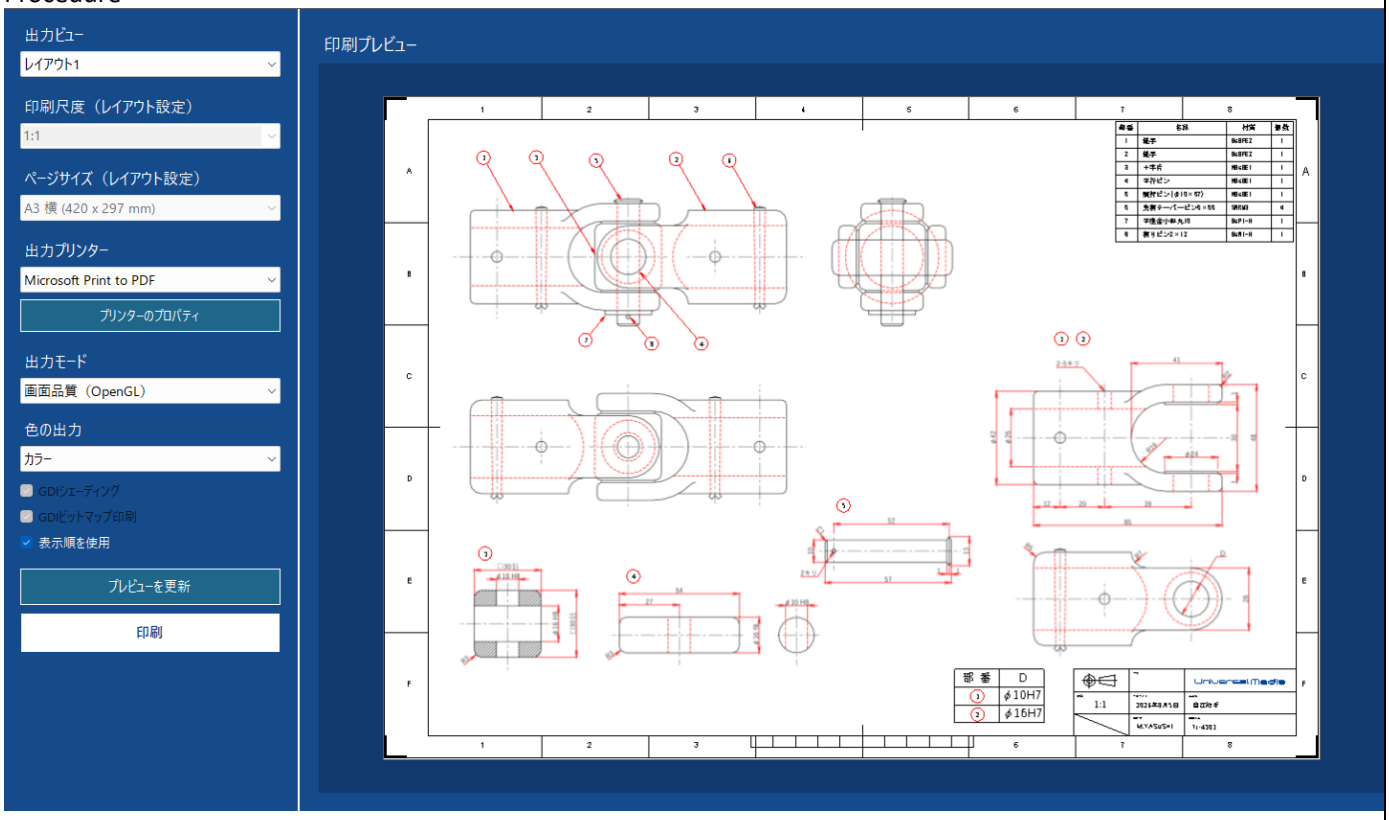
Procedure
The command closes the current drawing. If there are unsaved changes,
the confirmation dialog shown below appears.



Ribbon tab: File	Group: File Operations
Command: Print	Command ID: none (print screen)

Set the view, scale, paper, printer and output mode, then preview or print.

Procedure



Output view: Select the model or layout view to print.
Print scale: Set the scale when printing a model view.
For a layout view, the scale is fixed at 1:1.
Page size: Select the paper size when printing a model view.
A layout view uses the paper size defined in that layout.
Output printer: Choose the printer. Use Printer properties to open the printer driver's settings.
Output mode: Set output image quality and choose OpenGL or GDI.

Color output: Choose color or monochrome printing.

Ribbon tab: Home	Group: Draw
 Command: Polyline	Command ID: Constr.Polyline

Create a polyline through consecutively specified points.

Procedure

1. Left-click the start point.
2. Specify subsequent vertices in order. Line segments connect the points.
3. After the last point, check whether the shape should be open or closed and confirm.

Properties Explorer

Points: Specify the position of each vertex.

Length: Optionally specify the next segment's length.

End condition: Choose Close to connect the last point to the first.

Tips

Unlike continuous individual lines, the connected segments are treated as one polyline.

Ribbon tab: Home	Group: Draw
 Command: Construction line Cross	Command ID: Constr.DraftLine.Cross

Create horizontal and vertical construction lines through a point on the current workplane.

Procedure
Prompt

① Specify a point for the construction lines.
Left-click the reference point through which the construction lines pass.

Properties Explorer
Through point
You can enter X, Y and Z coordinates.

Ribbon tab: Home	Group: Draw
Command: Construction line Horizontal	Command ID: Constr.DraftLine.Horizontal

Create a horizontal construction line through a point on the current workplane.

Procedure
Prompt

① Specify a point for the construction lines.
Left-click the reference point through which the construction lines pass.

Properties Explorer
Through point
You can enter X, Y and Z coordinates.

Ribbon tab: Home	Group: Draw
 Command: Construction line Vertical	Command ID: Constr.DraftLine.Vertical

Create a vertical construction line through a point on the current workplane.

Procedure
Prompt

① Specify a point for the construction lines.
Left-click the reference point through which the construction lines pass.

Properties Explorer
Through point
You can enter X, Y and Z coordinates.

Ribbon tab: Home	Group: Draw
 Command: Construction line Two-point angle	Command ID: Constr.DraftLine.TwoPointAngle

Use two points to define the angle and create a construction line in that direction.

Procedure
Prompt

① Specify the first point for the angled construction line.
Left-click the first point defining the angled construction line.

② Specify the second point to define the angle.
Enter the second point determining the construction line's direction.

Properties Explorer
First point
You can enter X, Y and Z coordinates.

Second point
You can enter X, Y and Z coordinates.

Ribbon tab: Home	Group: Draw
 Command: Construction line Angle	Command ID: Constr.DraftLine.Angle

Create successive construction lines through specified points at an angle relative to the UCS.

Procedure
Prompt

① Enter the angle relative to the UCS.
Enter the angle in Properties Explorer.

② Specify a point for the construction lines.
Specify the point through which to insert the construction line.

Properties Explorer
Angle
Enter the angle.

Through point
You can specify X, Y and Z.

Ribbon tab: Home	Group: Draw
 Command: Construction line Relative angle	Command ID: Constr.DraftLine.RelativeAngle

Create successive construction lines through specified points at an angle relative to a reference object.

Procedure
Prompt

① Select the reference object for the relative angle.
Select the reference line.

② Enter the angle relative to the selected object.
Enter the angle in Properties Explorer.

③ Specify a point for the construction lines.
Specify the point through which to insert the construction line.

Properties Explorer
Angle
Enter the angle.

Through point
You can specify X, Y and Z.

Ribbon tab: Home	Group: Draw
 Command: Construction line Point and tangent	Command ID: Constr.DraftLine.PointTangent

Create successive construction lines tangent to a selected object from a specified point.

Procedure

1. Specify the point through which the construction line passes.
2. Select the tangent object, such as a circle or arc.
3. Check the side showing the required tangent and confirm. Continue to create another tangent if needed.

Properties Explorer

Start point: The point through which the tangent construction line passes. Coordinates can also be entered.

Tangent object: Select the object to which the line is tangent.

Tips

Creation is impossible if no tangent exists from the point, for example from inside a circle.

Ribbon tab: Home	Group: Draw
 Command: Construction line Two tangents	Command ID: Constr.DraftLine.TwoTangents

Create successive construction lines tangent to two selected objects.

Procedure

1. Select the first tangent object.
2. Select the second tangent object.
3. Inspect and confirm the common tangent. Change the pick location if a different candidate is shown.

Properties Explorer

Tangent object: The first tangent target.

Second tangent object: The second tangent target.

Tips

Depending on the objects, multiple internal or external tangent solutions may exist.

Ribbon tab: Home	Group: Draw
 Command: Construction line Fixed reference offset	Command ID: Constr.DraftLine.FixedReferenceParallel

Select a reference line and create successive parallel construction lines at entered distances from it.

Procedure

1. Select the reference object for the parallel lines.
2. Enter the distance from the reference, check the side and confirm.
3. Enter another distance to create the next line at that distance from the original reference.

Properties Explorer

Reference line: The object used as the fixed parallel-line reference.

Distance: Distance from the fixed reference line.

Tips

For example, entering 10 and then 20 creates lines at 10 and 20 from the original reference.

Ribbon tab: Home	Group: Draw
 Command: Construction line Moving reference offset	Command ID: Constr.DraftLine.MovingReferenceParallel

Select a reference and create a parallel construction line at the entered distance. Each new line becomes the next reference.

Procedure

1. Select the reference object for the parallel lines.

2. Enter a distance and create the construction line.

3. Enter the next distance. The line just created is now the reference.

Properties Explorer

Reference line: The initial reference object.

Distance: Distance from the current reference to the next construction line.

Tips

Entering 10 and then 20 on the same side creates lines at 10 and 30 from the original reference.

Ribbon tab: Home	Group: Draw
 Command: Construction line Delete all	Command ID: Constr.DraftLine.DeleteAll

Delete all construction lines.

Procedure

1. Check whether any construction lines need to be retained.

2. Run the command to delete all construction lines together.

Tips

To hide them temporarily, use Show or hide instead. Hiding and deletion are different operations.

Ribbon tab: Home	Group: Draw
 Command: Construction line Show or hide	Command ID: Constr.DraftLine.ToggleVisible

Show or hide all construction lines.

Procedure

1. Run the command to toggle construction-line visibility.

2. Run it again to restore the previous visibility state.

Tips

Hidden construction lines remain in the drawing. Use this to inspect finished geometry without its construction aids.

Ribbon tab: Home	Group: Draw
 Command: Line Single line	Command ID: Constr.Line.TwoPoints

Line: Specify a start point and endpoint.

Procedure

1. Left-click the start point or enter its coordinates.

2. Move the mouse and specify the endpoint to create the line between them.

Properties Explorer

Start / End: Coordinates of the line's endpoints.

Length / Direction: Define the line numerically.

Tips

Check the workplane and snap target before specifying points.

Ribbon tab: Home	Group: Draw
 Command: Line Continuous lines	Command ID: Constr.Line.RPoint

Start the continuous-line command.

Procedure

1. Specify the first point.

2. Specify the next point to create a segment. Its endpoint becomes the next segment's start.

3. Continue entering segments and press Esc to finish.

Properties Explorer

Points: Specify successive segment connection points.

Length: Optionally set the next segment's length.

Tips

Each segment is an independent line. Use Polyline to treat the whole chain as one object.

Ribbon tab: Home	Group: Draw
 Command: Line Length and angle	Command ID: Constr.Line.PointLengthAngle

Line: Specify a start point, length and direction.

Procedure

1. Specify the line's start point.
2. Enter its length and direction.
3. Check the direction in the preview and confirm.

Properties Explorer

Start point: The reference point of the line.

Length: Enter the line length.

Direction: Specify the line direction on the workplane.

Ribbon tab: Home	Group: Draw
 Command: Line Endpoint and tangent	Command ID: Constr.Line.PointTangent

Line: Specify a start point and tangent object.

Procedure

1. Specify the line's start point.
2. Select the tangent object, such as a circle or arc.
3. Pick near the required tangent, inspect the preview and confirm.

Properties Explorer

Start point: One endpoint of the tangent line.

Tangent object: The object touched by the other endpoint.

Tips

The tangent point is calculated geometrically. A point inside a circle cannot define a tangent to that circle.

Ribbon tab: Home	Group: Draw
 Command: Line Two tangents	Command ID: Constr.Line.TwoTangents

Line: Specify two tangent objects.

Procedure

1. Select the first tangent object.
2. Select the second tangent object.
3. Check the required tangent candidate and confirm.

Properties Explorer

Tangent object: The first target.

Second tangent object: The second target.

Tips

The pick location may affect the selected solution. Pick near the side where tangency is required.

Ribbon tab: Home	Group: Draw
 Command: Line Tangent and angle	Command ID: Constr.Line.TangentAngle

Line: Specify a direction and tangent object.

Procedure

1. Select a circle, arc or other tangent object.
2. Enter the line direction and length.
3. Check where the tangent point lies along the line and confirm.

Properties Explorer

Tangent object: The object from which to calculate tangency.

Direction / Length: Specify the tangent line's direction and length.

Position: Place the tangent point at the start or center of the line.

Ribbon tab: Home	Group: Draw
 Command: Line Perpendicular bisector	Command ID: Constr.Line.MiddlePerp

Line: Draw the perpendicular bisector of a segment.

Procedure

1. Select the segment to bisect.
2. Enter the length of the new line.
3. Check the direction through the segment midpoint and confirm.

Properties Explorer

Reference object: The segment to bisect.

Length: Length of the new perpendicular line.

Ribbon tab: Home	Group: Draw
 Command: Line Angle bisector	Command ID: Constr.Line.BisectAngle

Line: Draw an angle bisector.

Procedure

1. Select the first line defining the angle.

2. Select the second line.

3. Check the angle to bisect and line length, then confirm.

Properties Explorer

Angle-bisector objects: The two objects defining the angle.

Length: Enter the bisector length.

Tips

Check the preview to distinguish which of the two angles is being bisected.

Ribbon tab: Home	Group: Draw
 Command: Line Parallel by distance	Command ID: Constr.Line.ParallelDist

Line: Draw a parallel line at a specified distance from a segment.

Procedure

1. Select the reference line to copy in parallel.
2. Enter the distance and specify the side.
3. Check the spacing from the reference and confirm.

Properties Explorer

Reference line: The source segment.

Distance: Perpendicular distance from the reference.

Tips

For an inclined line, spacing is still measured normal to the reference, not horizontally or vertically.

Ribbon tab: Home	Group: Draw
 Command: Line Parallel through point	Command ID: Constr.Line.ParallelPoint

Line: Draw a parallel line through a point.

Procedure

1. Select the reference segment.
2. Specify a point through which the new segment must pass.
3. Check the point and parallel direction, then confirm.

Properties Explorer

Reference line: Defines direction and length.

Point: A point through which the parallel line passes.

Ribbon tab: Home	Group: Draw
 Command: Line Perpendicular through point	Command ID: Constr.Line.PointPerp

Line: Draw a line perpendicular to another object.

Procedure

1. Specify the start point of the perpendicular.
2. Select the reference object.
3. Check the perpendicular foot and line, then confirm.

Properties Explorer

Start point: Start of the perpendicular line.

Reference object: The line or curve onto which the perpendicular is dropped.

Tips

For a curve, pick near the desired perpendicular foot and inspect the candidate.

Ribbon tab: Home	Group: Draw
 Command: Line Multiline	Command ID: Constr.MultiLine

Create two to four continuous parallel lines from a center path.

Procedure

1. Set the number of lines and each offset from the center path.

2. Specify the center path's start and subsequent points.

3. If needed, enable Close, then inspect the shape and execute.

Properties Explorer

Count: Specify two to four lines.

Offsets: Set each line's position relative to the center path.

Close: Connect the end to the start.

Location: Home	Group: Draw
Line Circle centerline	Command ID: Constr.Line.CenterMark

Overview

Create crossed centerlines through the center of a circle or arc. Supports 2D drawing, circular 3D edges and projected layout views.

Procedure

1. Choose Home > Line drop-down > Circle centerline.

2. Click the circle or arc. In 3D, select a circular edge; in a layout, select a circle within the intended view.

3. Set Extension length if needed and confirm with the check mark.

Inputs and Result

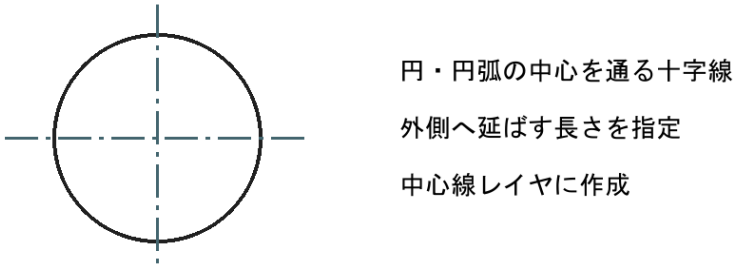
Extension length is the distance beyond the circle. It uses model lengths in a model view and paper millimeters in a layout.

Tips

Centerlines use the Centerline layer with ByLayer color, line pattern and width. Create the layer if absent; otherwise use its existing settings.

Associativity

Centerlines created from 3D edges or projected layout geometry retain source references. Those created from independent 2D circles are ordinary lines and do not follow circle edits automatically. Check positions after major source changes.



Illustration

Location: Home	Group: Draw
Line Centerline between two lines	Command ID: Constr.Line.CenterBetween

Overview

Create a centerline midway between two parallel lines. Select two straight objects, not two arbitrary points.

Procedure

1. Choose Home > Line drop-down > Centerline between two lines.

2. Click the first straight line, then the parallel second line.

3. Set Extension length, inspect the middle line and confirm with the check mark.

Selecting Targets

Select independent segments in 2D or straight model edges in 3D. Do not mix a 2D line and a 3D edge. In a layout, select both lines within the same view.

Inputs and Result

The centerline lies midway between the lines and extends past their ends by the specified amount. Layout extension length is measured in paper millimeters.

Tips

Nonparallel lines or selecting the same line twice cannot produce a centerline. Layer and association handling are the same as for Circle centerline.



Illustration

Ribbon tab: Home	Group: Draw
 Command: Circle Center and point	Command ID: Constr.Circle.CenterPoint

Circle: Specify its center and a point on the circumference.

Procedure

1. Specify the circle center.
2. Move the pointer to the intended circumference point.
3. Check the radius and click to create the circle.

Properties Explorer

Center: Circle-center coordinates.

Point on circle: Its distance from the center defines the radius.

Radius: Check the circle size.

Ribbon tab: Home	Group: Draw
 Command: Circle Center and radius	Command ID: Constr.Circle.CenterRadius

Circle: Specify center and radius.

Procedure

1. Specify the circle center.
2. Enter the radius, or use the diameter field for diameter input.
3. Check the center and size, then confirm.

Properties Explorer

Center: Circle-center coordinates.

Radius: Distance from the center to the circumference.

Diameter: Twice the radius. Do not confuse these values.

Tips

A radius of 10 gives a diameter of 20.

Ribbon tab: Home	Group: Draw
 Command: Circle Two points	Command ID: Constr.Circle.2Points

Circle: Specify two points.

Procedure

1. Specify one end of the diameter.
2. Specify the opposite end.
3. The circle is created with those points as diameter endpoints.

Properties Explorer

First / Second circle point: The diameter endpoints.

Center / Radius: Inspect the values calculated from the two points.

Tips

The center is the midpoint and the radius is half the point-to-point distance.

Ribbon tab: Home	Group: Draw
 Command: Circle Three points	Command ID: Constr.Circle.3Points

Circle: Specify three points.

Procedure

1. Specify the first and second circumference points.
2. Specify the third circumference point.
3. Inspect and confirm the circle through all three points.

Properties Explorer

First to third circle points: Three points on the circumference.

Center / Radius: Inspect the values calculated from the three points.

Tips

Three collinear points or coincident points do not define a circle.

Ribbon tab: Home	Group: Draw
 Command: Circle Two points and radius	Command ID: Constr.Circle.TwoPointsRadius

Circle: Specify two points and a radius.

Procedure

1. Specify the first and second points on the circle.
2. Enter the radius.
3. Choose which side of the two points contains the center and confirm.

Properties Explorer

First / Second circle point: Two circumference points.

Radius / Diameter: Use either value to specify size.

Tips

The radius must be at least half the distance between the points.

Ribbon tab: Home	Group: Draw
 Command: Circle Three tangents	Command ID: Constr.Circle.ThreeTangents

Circle: Draw a circle tangent to three objects.

Procedure

1. Select the first, second and third tangent objects in order.
2. Pick near the required tangency side and inspect the candidate.
3. Confirm the circle tangent to all three objects.

Properties Explorer

First to third tangent objects: The objects touched by the circle.

Radius: Inspect the size calculated from the constraints.

Tips

This command does not take a preset radius. The three tangency conditions determine it.

Ribbon tab: Home	Group: Draw
 Command: Circle Two tangents and point	Command ID: Constr.Circle.TwoTangentsPoint

Circle: Specify two tangent objects and a circumference point.

Procedure

1. Select the first and second tangent objects.
2. Specify a point on the new circle's circumference.
3. Inspect and confirm the solution tangent to both objects and passing through the point.

Properties Explorer

First / Second tangent object: The two tangency targets.

Point on circle: A circumference point, not the center.

Ribbon tab: Home	Group: Draw
 Command: Circle Two tangents and radius	Command ID: Constr.Circle.TwoTangentsRadius

Circle: Specify two tangent objects and a radius.

Procedure

1. Select the first and second tangent objects.
2. Enter the radius or diameter.
3. Check that the circle appears on the intended side and confirm.

Properties Explorer

First / Second tangent object: Objects touched by the circle.

Radius / Diameter: Size of the new circle.

Tips

If no circle of the specified radius can touch both objects, change the constraints.

Ribbon tab: Home	Group: Draw
 Command: Circle Tangent point and radius	Command ID: Constr.Circle.TangentPointRadius

Circle: Specify a tangent object, point and radius.

Procedure

1. Select the object to which the circle is tangent.
2. Specify a circumference point.
3. Enter the radius, inspect the desired candidate and confirm.

Properties Explorer

First tangent object: The object touched by the circle.

Point on circle: A point on the circumference.

Radius / Diameter: Specify circle size.

Tips

Point on circle does not mean the center point.

Location: Home	Group: Draw
Circle Tangent center locus and radius	Command ID: Constr.Circle.TangentCenterCurveRadius

Overview

Create a circle of the specified radius tangent to one object with its center on another. The center locus can be a line, circle, arc or other supported object.

Procedure

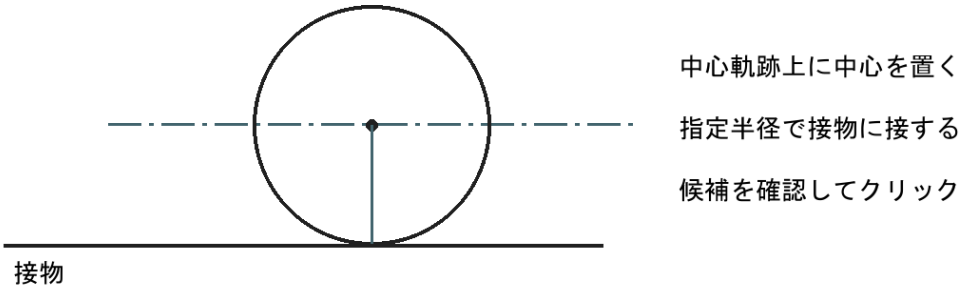
1. Choose Home > Circle drop-down > Tangent, center locus and radius.
2. Select the object that the new circle must touch.
3. Select the object on which its center must lie as the center locus.
4. Enter the radius and press Enter.
5. Move near the intended circle, inspect the candidate and click.

Inputs

Tangent object: The object to touch. Center locus: The object carrying the center. Radius: Circle size. Where multiple solutions exist, use the mouse position to choose.

Tips

Both objects must be coplanar. Tangency and center points must lie within their actual extents, not on extensions. If no solution exists, revise the radius or targets. Source objects remain unchanged.



Illustration

Ribbon tab: Home	Group: Draw
 Command: Arc Center and point	Command ID: Constr.Arc.CenterStartEnd

Arc: Specify center, start and end.

Procedure

1. Specify the arc center.
2. Specify its start point to set the radius.
3. Specify the end direction, check the sweep direction and confirm.

Properties Explorer

Center / Start / End: Define the arc's position and extent.

Direction: Set the sweep direction.

Radius / Start angle / End angle: Inspect or set size and angular range.

Ribbon tab: Home	Group: Draw
 Command: Arc Center and radius	Command ID: Constr.Arc.CenterRadius

Arc: Specify center, radius, start angle and end angle.

Procedure

1. Specify center and radius.
2. Enter start and end angles.
3. Check the sweep direction and confirm the arc.

Properties Explorer

Center: The arc center.

Radius / Diameter: Specify arc size.

Start angle / End angle / Direction: Define the arc range.

Tips

Inspect the entire preview to avoid selecting the long arc instead of the short one, or vice versa.

Ribbon tab: Home	Group: Draw
 Command: Arc Two points and radius	Command ID: Constr.Arc.2PointsRadius

Arc: Specify two points and a radius.

Procedure

1. Specify the first and second arc points.
2. Enter the radius.
3. Check which side the arc bulges toward and confirm.

Properties Explorer

First / Second circumference point: The arc endpoints.

Radius / Diameter: Specify arc size.

Center: Inspect the center of the chosen solution.

Tips

An arc cannot be created if its radius is less than half the endpoint distance.

Ribbon tab: Home	Group: Draw
 Command: Arc Three points	Command ID: Constr.Arc.3Points

Arc: Specify start, intermediate point and end.

Procedure

1. Specify the arc start.
2. Specify an intermediate point.
3. Specify the end to create an arc through all three points.

Properties Explorer

First to third circumference points: Start, intermediate point and end, in that order.

Center / Radius: Inspect values determined by the three points.

Tips

The side containing the intermediate point determines the arc's bulge.

Ribbon tab: Home	Group: Draw
 Command: Arc Three tangents	Command ID: Constr.Arc.ThreeTangents

Arc: Draw an arc tangent to three objects.

Procedure

1. Select the three tangent objects in order.
2. Check the tangency sides and arc range in the preview.
3. Confirm the desired candidate.

Properties Explorer

First to third tangent objects: Objects touched by the arc.

Radius: Inspect the size determined by tangency.

Tips

Different pick positions on the same objects may select different solutions.

Ribbon tab: Home	Group: Draw
 Command: Arc Two tangents and point	Command ID: Constr.Arc.TwoTangentsPoint

Arc: Specify two tangent objects and a circumference point.

Procedure

1. Select two tangent objects.

2. Specify a circumference point through which the arc passes.

3. Check the tangency sides and arc range, then confirm.

Properties Explorer

First / Second tangent object: Objects touched by the arc.

Point on circumference: A position through which the arc passes.

Ribbon tab: Home	Group: Draw
 Command: Arc Two tangents and radius	Command ID: Constr.Arc.TwoTangentsRadius

Arc: Specify two tangent objects and a radius.

Procedure

1. Select two tangent objects.
2. Enter the radius or diameter.
3. Check the tangent points and sweep direction, then confirm.

Properties Explorer

First / Second tangent object: Objects touched by the arc.

Radius / Diameter: Specify arc size.

Tips

To round a corner in existing lines, you can also use Modify > Fillet.

Ribbon tab: Home	Group: Draw
 Command: Arc Tangent point and radius	Command ID: Constr.Arc.TangentPointRadius

Arc: Specify a tangent object, point and radius.

Procedure

1. Select the object to which the arc is tangent.
2. Specify a point through which the arc passes.
3. Enter the radius, inspect the candidate and confirm.

Properties Explorer

First tangent object: The tangency target.

Point on circumference: A point on the arc.

Radius / Diameter: Specify size.

Ribbon tab: Home	Group: Draw
 Command: Rectangle Opposite corners	Command ID: Constr.Rect.2DiagonalPoints

Rectangle: Specify two opposite corners.

Procedure

1. Specify the first corner.

2. Move to and specify the opposite corner.

Properties Explorer

Start point: One corner of the rectangle.

Opposite corner: The diagonally opposite corner.

Tips

Sides align with the current workplane axes. Use an angle-based method for an inclined rectangle.

Ribbon tab: Home	Group: Draw
 Command: Rectangle Center width and height	Command ID: Constr.Rect.CenterWidthHeight

Rectangle: Specify center, width and height.

Procedure

1. Specify the rectangle center.
2. Specify a size point or enter width and height.
3. Check the symmetric shape about the center and confirm.

Properties Explorer

Center: Rectangle center.

Size point: Set width and height with the mouse.

Width / Height: Overall rectangle dimensions.

Tips

Width 40 and height 20 extend 20 left/right and 10 up/down from the center.

Ribbon tab: Home	Group: Draw
 Command: Rectangle Width height and angle	Command ID: Constr.Rect.PointWidthHeightAngle

Rectangle: Specify a point, width, height and direction.

Procedure

1. Specify the starting corner.
2. Enter width and height.
3. Specify the reference-side direction, check orientation and confirm.

Properties Explorer

Start point: The reference corner.

Width / Height: Dimensions of the two sides.

Direction: Angle of the width side.

Ribbon tab: Home	Group: Draw
 Command: Rectangle Two points and height	Command ID: Constr.Rect.2PointsHeight

Rectangle: Specify two reference points and height.

Procedure

1. Specify the start of the reference side.
2. Specify the second point on that side.
3. Enter the height, check the construction side and confirm.

Properties Explorer

Start / Second point: Define the reference side's position, length and direction.

Height: Dimension perpendicular to that side.

Ribbon tab: Home	Group: Draw
 Command: Parallelogram	Command ID: Constr.Rect.Parallelogram

Parallelogram: Specify three points.

Procedure

1. Specify the starting corner.
2. Specify the adjacent vertex in the primary direction.
3. Specify the secondary direction, inspect the shape formed by the two sides and confirm.

Properties Explorer

Start point: The shared corner of two sides.

Primary / Secondary direction: Orientations and lengths of adjacent sides.

Tips

Parallel directions cannot produce a shape with nonzero area.

Ribbon tab: Home	Group: Draw
 Command: Polygon	Command ID: Constr.Polygon

Create an inscribed or circumscribed polygon.

Procedure

1. Set the number of sides and construction method.
2. Specify center and radius.
3. Specify the reference angle and confirm.

Properties Explorer

Sides: Number of polygon sides.

Method: Choose Inscribed or Circumscribed.

Center / Radius / Angle: Set position, size and orientation.

Tips

An inscribed polygon has vertices on the reference circle; a circumscribed polygon has sides tangent to it. The same radius gives different overall sizes.

Ribbon tab: Home	Group: Draw
 Command: Ellipse Three points	Command ID: Constr.Ellipse.CenterRadius

Ellipse: Specify center and radii.

Procedure

1. Specify the ellipse center.
2. Specify the first vertex to define the major-axis side.
3. Specify the second vertex to define the minor-axis side and confirm.

Properties Explorer

Center: Ellipse-center coordinates.

First / Second vertex: Vertices on the two axes.

First / Second radius and ellipse angle: Specify axis radii and orientation.

Tips

Radius fields measure from center to vertex, not the full width.

Ribbon tab: Home	Group: Draw
 Command: Ellipse Two points and two directions	Command ID: Constr.Ellipse.2PointsDirections

Ellipse: Specify two points and two directions.

Procedure

1. Specify the first ellipse point and its tangent direction.
2. Specify the second point and its tangent direction.
3. Inspect and confirm the ellipse satisfying both points and directions.

Properties Explorer

First / Second point: Points on the ellipse.

First / Second direction: Tangent directions at those points.

Tips

These directions are tangents at the selected points, not the major and minor axes.

Ribbon tab: Home	Group: Draw
 Command: Elliptical arc Three points	Command ID: Constr.Ellipsearc.CenterRadius

Elliptical arc: Specify center and two radii.

Procedure

1. Define the ellipse with its center and first and second vertices.
2. Specify start and end angles.
3. Check the retained arc direction and confirm.

Properties Explorer

Center / First radius / Second radius: Ellipse position and size.

Ellipse angle: Overall ellipse orientation.

Start angle / End angle / Direction: Elliptical arc extent.

Ribbon tab: Home	Group: Draw
 Command: Elliptical arc Two points and two directions	Command ID: Constr.Ellipsearc.2PointsDirections

Elliptical arc: Specify two points and two directions.

Procedure

1. Specify the first point and its tangent direction.
2. Specify the second point and its tangent direction.
3. Check the arc direction connecting the two points and confirm.

Properties Explorer

First / Second point: Arc endpoints.

First / Second direction: Tangent directions at each endpoint.

Direction: Sweep direction of the elliptical arc.

Ribbon tab: Home	Group: Draw
 Command: Spline	Command ID: Constr.BSpline.Points

Create a NURBS spline.

Procedure

1. Specify successive points through which the spline passes.

2. Add the required points and inspect the curve.

3. Choose open or closed and execute.

Properties Explorer

Points: Add and inspect the points defining the curve.

Mode: Choose Open or Closed.

Tips

Too many points can make a curve oscillate. Use only the points needed to describe the shape.

Draw Clothoids

Choose Home > Draw > Spline drop-down > Clothoid. A clothoid is a planar curve that can be placed on any workplane.

Formula and Inputs

$A^2 = R \times L$. A is the clothoid parameter, L is arc length from the zero-curvature point KA, and R is curvature radius there. Use the same length unit for A, L and R.

1. Check the workplane and choose placement method, numerical definition method and left/right turning.

2. Specify the base point and a point defining direction.

3. Enter A and start/end distances, or A and start/end radii.

4. Check preview length and direction and confirm with the check mark.

Input Example	Values	Resulting curve
Full curve from KA	A = 100, start distance 0, end distance 100.	Length 100, end radius 100.
An intermediate segment	A = 100, start distance 20, end distance 100.	Length 80, start radius 500, end radius 100.
Zero in radius input	Start radius 0.	Infinite radius: the connection to a straight line.

Example lengths use mm when model units are mm. With meter display, the same numbers represent meters, so match the design dimensions.

Placement Notes

Base point and entry direction specifies KA and its tangent. In Start point and end direction, the second point also defines direction; it does not calculate a length reaching that point. A and the distance/radius inputs determine length.

After creation, edit KA position, A and start/end distances in properties. Invalid values are rejected and previous values retained.

Basic Clothoid Alignment and Cross Lines

Create a Clothoid Arc Clothoid Alignment from IP

Choose Spline drop-down > Basic clothoid (IP). Specify an incoming point, tangent intersection IP and outgoing point, then enter A and final radius R. This creates a symmetric basic alignment with the same A on both sides.

The three points define incoming and outgoing directions; existing straight lines are not trimmed automatically. If transition lengths are excessive for the deflection angle, reduce A or increase R. Collinear points, a 180-degree reversal and different A values on each side are unsupported.

Normals Tangents and Regular Cross Lines

Choose a mode in Clothoid normals, tangents and cross lines, then click the curve. Distance here is arc length from the displayed curve's start, not from KA.

Mode	Inputs
Normal at distance	Distance from start and left/right lengths.
Tangent at distance	Distance from start and forward/backward lengths.
Normal from external point	External point; the shortest normal is used when several solutions exist.
Regular cross lines	Start distance, spacing and left/right lengths.

For length 100, start 0, spacing 20 and 5 on each side, six width-10 cross lines are created at 0, 20, 40, 60, 80 and 100. No extra endpoint line is added if length is not an exact multiple of spacing. Limit: 2,000 lines per operation.

Saving and Re-editing

JDC retains dedicated clothoid objects. Standard XY-plane curves export as SXF clothoids; DWG/DXF use polyline approximations. Cross lines are independent segments and do not follow curve edits. Vertical alignments, superelevation and road sections are not generated automatically.

Ribbon tab: Home	Group: Draw
 Command: Helix	Command ID: Constr.Helix

Create a helical path.

Procedure

1. Specify the helix center.
2. Enter bottom radius, top radius, height and turn count.
3. Check rotation direction and shape, then confirm.

Properties Explorer

Bottom / Top radius: Radii at the start and end. Equal values give a constant diameter.

Height / Turns: Axial height and number of revolutions.

Smoothness per turn: Number of interpolation points. Larger values increase data size.

Rotation direction: Specified relative to the workplane normal.

Tips

This command creates a helix path. Use it in a separate operation to create solid wire or similar geometry.

Ribbon tab: Home	Group: Draw
 Command: Point	Command ID: Constr.Point

Create a simple point.

Procedure

1. Left-click the point location.

2. Continue placing points as required, then end input.

Properties Explorer

Point: You can enter X, Y and Z coordinates.

Tips

Appearance follows point-display settings. This command does not create a circle or hole.

Ribbon tab: Home	Group: Draw
 Command: Divide	Command ID: Constr.Divide

Divide a selected object into equal intervals and place points at the division positions.

Procedure

1. Select the line, arc or curve to divide.
2. Enter the division count.
3. Inspect the equally spaced points and confirm.

Properties Explorer

Object: The curve on which to calculate division positions.

Divisions: Number of equal intervals.

Tips

This places points; it does not split the source curve.

Ribbon tab: Home	Group: Draw
 Command: Hatch	Command ID: Constr.Hatch.WithHoles

Create a hatch with internal holes.

Procedure

1. Prepare a closed outer boundary and closed internal hole boundaries.
2. Specify a point inside the region to hatch.
3. Check that holes are excluded, verify the hatch style and confirm.

Properties Explorer

Interior point: A point within the filled region.

Hatch style: Set the line pattern and related appearance.

Tips

Gaps can prevent region detection. Use Check closed geometry to inspect endpoints.

Ribbon tab: Home	Group: Draw
 Command: Simple hatch	Command ID: Constr.Hatch.Inside

Create a hatch using the outermost boundary.

Procedure

1. Prepare a closed region.
2. Specify a point inside it.
3. Inspect the filled area and style, then confirm.

Properties Explorer

Interior point: Identify the hatch region.

Hatch style: Set the displayed pattern.

Tips

If internal boundaries must remain empty, use the standard Hatch command and check hole handling.

Ribbon tab: Home	Group: Draw
Command: Boundary	Command ID: Constr.Boundary

Specify an interior point to create a closed boundary path.

Procedure

1. Prepare a region enclosed by lines and arcs.
2. Specify a point inside the region whose boundary is required.
3. Inspect the detected outline and confirm.

Properties Explorer

Interior point: A point inside the closed region.

Tips

Creates a new closed path. To connect the original lines directly, use Join.

Ribbon tab: Home	Group: Draw
 Command: Check closed geometry	Command ID: JDesign.CheckClosedContours

Check whether selected lines, arcs, polylines, paths and splines form closed geometry, and preview open endpoints.

Procedure

1. Select the lines, arcs, polylines or other objects to check.
2. Run the command and inspect the result.
3. If open endpoints are displayed, zoom to them and repair the geometry.

Tips

The check does not repair gaps. Align the endpoints, recheck, then use the geometry for extrusion or hatching.

Ribbon tab: Home	Group: Draw
 Command: Image	Command ID: Constr.Picture.RefPointWidthHeight

Image: Specify anchor, width and height.

Procedure

1. Choose an image file.
2. Specify its anchor position.
3. Set width, height and angle, then confirm.

Properties Explorer

Image / Position: Choose the file and placement position.

Width / Height / Baseline angle: Set size and orientation.

Keep aspect ratio: Prevent unwanted image distortion.

Tips

The image is not converted into CAD lines, circles or other geometry.

Ribbon tab: Home	Group: Draw
Command: Image by two points	Command ID: Constr.Picture.RefPoint2Directions

Image: Specify anchor, width direction and height direction.

Procedure

1. Choose the image and specify its anchor.
2. Specify width and height directions on screen.
3. Check size and orientation, then confirm.

Properties Explorer

Position: Image anchor point.

Baseline / Left-edge direction: Two placement directions.

Width / Height / Scale: Specify placement size.

Keep aspect ratio / Rectangular: Preserve the image's shape.

Ribbon tab: Home	Group: Modeling
 Command: Solid Box	Command ID: Constr.Box

Create a box with length, width and height, optionally along arbitrary directions.

Procedure

1. Specify the box's starting corner.
2. Enter length, width and height. For an arbitrary orientation, specify each axis direction too.
3. Check insertion mode and shape, then execute.

Properties Explorer

Start point: Reference corner of the box.

Length / Width / Height: Dimensions in three directions.

X / Y / Z direction: Directions of the box edges.

Insertion mode: Choose New, Union, Subtract or another mode.

Ribbon tab: Home	Group: Modeling
 Command: Solid Cylinder	Command ID: Constr.Cylinder

Create a cylinder using an axis and radius, or radius and height.

Procedure

1. Specify the base center.
2. Specify radius and height. For an inclined cylinder, use a point on the axis to set direction and height.
3. Check insertion mode and execute.

Properties Explorer

Center: Cylinder reference position.

Point on axis: Defines axis direction and length.

Radius / Height: Cylinder size.

Insertion mode: Add to or cut from an existing solid, or create separately.

Ribbon tab: Home	Group: Modeling
 Command: Solid Cone	Command ID: Constr.Cone

Create a cone using an axis and radii, or radii and height.

Procedure

1. Specify the base center.

2. Enter first radius, second radius and height. Specify a point on the axis if needed.

3. Check shape and insertion mode, then execute.

Properties Explorer

Center / Point on axis: Cone position and axis.

First / Second radius: End radii. The second radius may be zero.

Height: Axial length.

Insertion mode: Create a new object or operate on an existing solid.

Tips

A second radius of zero creates a pointed cone; a nonzero value creates a conical frustum.

Ribbon tab: Home	Group: Modeling
 Command: Solid Sphere	Command ID: Constr.Sphere

Create a sphere from a center and radius.

Procedure

1. Specify the sphere center.
2. Enter the radius.
3. Check insertion mode and execute.

Properties Explorer

Center: Sphere-center coordinates.

Radius: Distance from center to the spherical surface.

Insertion mode: New, Union, Subtract or another mode.

Ribbon tab: Home	Group: Modeling
 Command: Torus	Command ID: Constr.Torus

Create a torus with center, major radius and minor radius. A point on the axis can also be specified.

Procedure

1. Specify the torus center.
2. Enter major and minor radii.
3. Specify axis direction if needed, check shape and insertion mode, then execute.

Properties Explorer

Center / Point on axis: Torus position and axial direction.

Major radius: Distance from torus center to tube-section center.

Minor radius: Radius of the tube cross-section.

Insertion mode: Boolean operation with existing solids.

Tips

For a conventional ring, outer radius equals major plus minor radius; inner radius equals major minus minor radius.

Ribbon tab: Home	Group: Modeling
 Command: Extrude	Command ID: Constr.Solid.FaceExtrude

Extrude a face or closed curve to create a solid.

Procedure

1. Select a closed profile or face. A region with holes can also be selected using Interior point.

2. Choose extrusion mode and distribution, then specify height or the destination plane.

3. Choose New, Union, Subtract or another insertion mode, check direction and execute.

Properties Explorer

Height / Distribution: Centered distribution assigns half the entered height to each side.

Target plane object: Destination when extruding up to a face.

Taper angle: Positive values narrow the far end; negative values widen it.

Extrusion plane / Direction: Set the profile reference plane and extrusion direction.

Offset distance / Direction: Shift the extrusion start position.

Extrusion path: Specify a curve for path-based extrusion.

Insertion mode: Union joins material; Subtract cuts existing solids.

Tips

Example: Centered height 140 extrudes 70 on each side of the profile. For subtraction, ensure the cutting shape reaches the target.

Ribbon tab: Home	Group: Modeling
 Command: Revolve	Command ID: Constr.Solid.FaceRotate

Revolve a face or closed curve to create a solid.

Procedure

1. Select a closed profile or face to revolve.
2. Select the rotation axis, or enter a point on the axis and its direction.
3. Set angle, rotation direction and distribution.
4. Check insertion mode and preview, then execute.

Properties Explorer

Axis / Point on axis / Axis direction: Define the rotation line.

Angle / Rotation direction: Specify sweep extent and direction.

Revolution mode: Choose an angle or a target plane.

Distribution / Offset: Specify sweep distribution and starting position.

Insertion mode: Choose New, Union, Subtract or another mode.

Tips

For subtraction, the revolved body must intersect the target solid. Confirm both profile and axis inputs before executing.

Ribbon tab: Home	Group: Modeling
 Command: Loft	Command ID: Constr.Solid.Loft

Create a loft solid through closed profiles in selection order.

Procedure

1. Select at least two closed planar profiles in connection order.

2. Once all profiles are selected, set insertion mode.

3. Check twisting and profile order in the preview, then press Enter.

Properties Explorer

Profiles: Selection order determines loft connection order.

Insertion mode: New, Union or Subtract.

Tips

Click a selected profile again to deselect it. Use closed profiles for a solid.

Ribbon tab: Home	Group: Modeling
 Command: Path loft	Command ID: Constr.Solid.PathLoft

Loft closed profiles along a center path.

Procedure

1. Select at least two closed planar profiles in order along the path.
2. Press Enter to finish profile selection, then select an open center path.
3. Check insertion mode and shape, then press Enter to execute.

Properties Explorer

Profiles: Specify the sequence, including intermediate shapes.

Center path: The curve followed by profile positions and orientations.

Insertion mode: Choose New, Union or Subtract.

Tips

For preselection, select the profiles in order and the center path last.

Ribbon tab: Home	Group: Modeling
 Command: Extrude	Command ID: Constr.Face.ObjectExtrude

Extrude a path, curve or connected curves to create a surface.

Procedure

1. Select the line, arc or path to extrude.

2. Specify height and distribution, or a target plane.

3. Check direction and surface shape, then execute.

Properties Explorer

Object: Curve to extrude into a surface.

Height / Distribution: Extrusion length and allocation to each side.

Target plane object: Specify when extruding up to a face.

Extrusion plane / Direction / Offset: Set the reference and start position.

Extrusion path: Specify when following a curve.

Tips

This creates surfaces. Use solid extrusion when a body with volume is required.

Ribbon tab: Home	Group: Modeling
 Command: Revolve	Command ID: Constr.Face.ObjectRotate

Revolve a path, curve or connected curves to create a surface.

Procedure

1. Select the curve or path to revolve.
2. Select an axis or specify a point on it and its direction.
3. Set angle and direction, inspect the preview and execute.

Properties Explorer

Object: Curve to revolve.

Axis / Point on axis / Axis direction: Define the rotation axis.

Angle / Direction / Distribution: Define the swept range.

Target plane / Offset: Set the destination or starting angle where required.

Tips

The result is a revolved surface. Distinguish this from solid revolution.

Ribbon tab: Home	Group: Modeling
 Command: Planar surface	Command ID: Constr.Face.FromObject

Convert a curve, connected curves or path into a planar face.

Procedure

1. Specify a closed curve or connected outline lying in one plane.
2. To create a region with holes, specify an interior point.
3. Inspect the face region and confirm.

Properties Explorer

Curve: Closed outline forming the face boundary.

Interior point: Identify a region, including holes, from a point inside it.

Tips

The face has no thickness. Extrude it to create a thick body.

Ribbon tab: Home	Group: Modeling
 Command: Loft	Command ID: Constr.Face.Loft

Create a loft surface through profiles in selection order.

Procedure

1. Select at least two profiles in connection order.
2. Check surface twist and connection direction.
3. Press Enter to confirm the loft surface.

Properties Explorer

Profiles: Use all open or all closed profiles.

Tips

Do not mix open and closed profiles. This command creates surfaces even when closed profiles are used.

Ribbon tab: Home	Group: Modeling
 Command: Path loft	Command ID: Constr.Face.PathLoft

Loft profiles along a center path.

Procedure

1. Select at least two profiles in order along the path.

2. Press Enter to finish profile selection, then select an open center path.

3. Inspect the changing surface along the path and press Enter.

Properties Explorer

Profiles: Use all open or all closed profiles.

Center path: Curve defining profile positions and orientations.

Tips

Use one continuous center path and check profile selection order.

Ribbon tab: Home	Group: Modeling
 Command: Network	Command ID: Constr.Face.NetworkSurface

Create a surface from two intersecting curve families.

Procedure

1. Select at least two curves in the first direction.
2. Press Enter to confirm the first family.
3. Select at least two intersecting curves in the second direction.
4. Inspect the preview and press Enter.

Properties Explorer

First curve family: Curves defining one surface direction.

Second curve family: Curves crossing the first family.

Tips

Arrange the curves to intersect as a grid. Curves that only appear to overlap without intersecting cannot define the surface.

Ribbon tab: Home	Group: Modeling
 Command: XY workplane	Command ID: DrawingPlane.StandardXY

Set the XY plane as the current workplane.

Procedure

1. Run the command to select the WCS XY plane.

2. Check the UCS axes and workplane orientation before drawing.

Tips

This changes the reference plane for subsequent drawing; it does not rotate the existing model.

Ribbon tab: Home	Group: Modeling
 Command: Offset XY workplane	Command ID: DrawingPlane.StandardXY.Offset

Set an offset XY plane as the current workplane.

Procedure

1. Run the XY workplane offset command.
2. Enter the normal offset distance.
3. Check the displacement side and confirm.

Properties Explorer

Normal offset: Signed distance along the reference plane's normal.

Tips

The sign changes the displacement side, which need not coincide with screen up/down.

Ribbon tab: Home	Group: Modeling
 Command: XZ workplane	Command ID: DrawingPlane.StandardXZ

Set the XZ plane as the current workplane.

Procedure

1. Run the command to select the WCS XZ plane.

2. Check the UCS axes and workplane orientation before drawing.

Tips

This changes the reference plane for subsequent drawing; it does not rotate the existing model.

Ribbon tab: Home	Group: Modeling
 Command: Offset XZ workplane	Command ID: DrawingPlane.StandardXZ.Offset

Set an offset XZ plane as the current workplane.

Procedure

1. Run the XZ workplane offset command.
2. Enter the normal offset distance.
3. Check the displacement side and confirm.

Properties Explorer

Normal offset: Signed distance along the reference plane's normal.

Tips

The sign changes the displacement side, which need not coincide with screen up/down.

Ribbon tab: Home	Group: Modeling
 Command: YZ workplane	Command ID: DrawingPlane.StandardYZ

Set the YZ plane as the current workplane.

Procedure

1. Run the command to select the WCS YZ plane.

2. Check the UCS axes and workplane orientation before drawing.

Tips

This changes the reference plane for subsequent drawing; it does not rotate the existing model.

Ribbon tab: Home	Group: Modeling
 Command: Offset YZ workplane	Command ID: DrawingPlane.StandardYZ.Offset

Set an offset YZ plane as the current workplane.

Procedure

1. Run the YZ workplane offset command.
2. Enter the normal offset distance.
3. Check the displacement side and confirm.

Properties Explorer

Normal offset: Signed distance along the reference plane's normal.

Tips

The sign changes the displacement side, which need not coincide with screen up/down.

Ribbon tab: Home	Group: Modeling
 Command: Offset workplane	Command ID: DrawingPlane.Current.Offset

Move the current workplane a specified distance along its normal.

Procedure

1. Check the current workplane.
2. Enter the displacement in Normal offset.
3. Inspect the moved plane and confirm.

Properties Explorer

Normal offset: Signed distance from the current workplane.

Tips

The plane moves perpendicular to itself, not along its local X or Y direction.

Ribbon tab: Home	Group: Modeling
 Command: Face on model	Command ID: DrawingPlane.Tangential

Select a planar face on the model.

Procedure

1. Select the model face to use as a reference.
2. Check the workplane and UCS orientation along the selected face.
3. Start drawing on that plane.

Tips

Select a planar face; a curved surface cannot serve as the same plane. Change viewpoint if the face is difficult to select.

Ribbon tab: Home	Group: Modeling
 Command: Face edge and angle	Command ID: DrawingPlane.AngleToFace

Create a workplane at a specified angle about a straight edge on a planar face.

Procedure

1. Select the reference planar face.

2. Select a straight edge on that face as the rotation axis.

3. Enter the angle, inspect the inclined workplane and confirm.

Properties Explorer

Face: Reference plane before inclination.

Edge: Straight axis about which the plane rotates.

Angle: Inclination from the reference face.

Tips

Select a straight edge on the face, not an arc edge.

Ribbon tab: Home	Group: Modeling
 Command: Path and curve plane	Command ID: DrawingPlane.OfCurve

Set the plane containing a path or planar curve as the workplane.

Procedure

1. Select a planar circle, arc or path.

2. Check its containing plane and use that as the workplane.

Tips

A spatially twisted curve cannot define one plane. Select a planar curve.

Ribbon tab: Home	Group: Modeling
 Command: Workplane by three points	Command ID: DrawingPlane.Three.Points

Define a workplane with three points.

Procedure

1. Specify the first point as the workplane origin.

2. Specify the second point to define X direction.

3. Specify the third point to define the plane, inspect the UCS and confirm.

Properties Explorer

Three points: Define the origin, axis direction and plane orientation in order.

Tips

Collinear points do not define a plane. Point order affects orientation.

Ribbon tab: Home	Group: Modeling
 Command: Workplane perpendicular to path	Command ID: DrawingPlane.Path.Perpendicular

Create a workplane normal to the tangent at a selected path or curve endpoint.

Procedure

1. Select the reference path or curve.
2. Specify the endpoint where the plane is required.
3. Check that the plane is normal to the endpoint tangent.

Tips

Use this to draw a profile for path extrusion. Choosing the opposite endpoint changes position and orientation.

Ribbon tab: Home	Group: Modeling
 Command: WCS X angle	Command ID: DrawingPlane.WcsXAngle

Create a workplane rotated by a specified angle about the WCS X axis.

Procedure

1. Run the angle command based on WCS X.
2. Enter the angle.
3. Inspect the rotated plane and UCS orientation, then confirm.

Properties Explorer

Angle: Rotation about WCS X.

Tips

WCS is the drawing's fixed coordinate system; UCS belongs to the current workplane. After tilting the workplane, identically named axes may point in different directions.

Ribbon tab: Home	Group: Modeling
 Command: WCS Y angle	Command ID: DrawingPlane.WcsYAngle

Create a workplane rotated by a specified angle about WCS Y.

Procedure

1. Run the angle command based on WCS Y.
2. Enter the angle.
3. Inspect the rotated plane and UCS orientation, then confirm.

Properties Explorer

Angle: Rotation about WCS Y.

Tips

WCS is the drawing's fixed coordinate system; UCS belongs to the current workplane. After tilting the workplane, identically named axes may point in different directions.

Ribbon tab: Home	Group: Modeling
 Command: WCS Z angle	Command ID: DrawingPlane.WcsZAngle

Create a workplane rotated by a specified angle about WCS Z.

Procedure

1. Run the angle command based on WCS Z.
2. Enter the angle.
3. Inspect the rotated plane and UCS orientation, then confirm.

Properties Explorer

Angle: Rotation about WCS Z.

Tips

WCS is the drawing's fixed coordinate system; UCS belongs to the current workplane. After tilting the workplane, identically named axes may point in different directions.

Ribbon tab: Home	Group: Modeling
 Command: UCS X angle	Command ID: DrawingPlane.UcsXAngle

Create a workplane rotated by a specified angle about the current UCS X axis.

Procedure

1. Run the angle command based on the current workplane's X axis.
2. Enter the angle.
3. Inspect the rotated plane and UCS orientation, then confirm.

Properties Explorer

Angle: Rotation about UCS X.

Tips

WCS is the drawing's fixed coordinate system; UCS belongs to the current workplane. After tilting the workplane, identically named axes may point in different directions.

Ribbon tab: Home	Group: Modeling
 Command: UCS Y angle	Command ID: DrawingPlane.UcsYAngle

Create a workplane rotated by a specified angle about the current UCS Y axis.

Procedure

1. Run the angle command based on the current workplane's Y axis.
2. Enter the angle.
3. Inspect the rotated plane and UCS orientation, then confirm.

Properties Explorer

Angle: Rotation about UCS Y.

Tips

WCS is the drawing's fixed coordinate system; UCS belongs to the current workplane. After tilting the workplane, identically named axes may point in different directions.

Ribbon tab: Home	Group: Modeling
 Command: UCS Z angle	Command ID: DrawingPlane.UcsZAngle

Create a workplane rotated by a specified angle about the current UCS Z axis.

Procedure

1. Run the angle command based on the current workplane's Z axis.
2. Enter the angle.
3. Inspect the rotated plane and UCS orientation, then confirm.

Properties Explorer

Angle: Rotation about UCS Z.

Tips

WCS is the drawing's fixed coordinate system; UCS belongs to the current workplane. After tilting the workplane, identically named axes may point in different directions.

Ribbon tab: Home	Group: Modify
 Command: Move	Command ID: JDesign.Modify.Move

Move selected objects.

Procedure

1. Select the objects and start the command.

2. Specify the source reference point.

3. Specify the destination point. The objects move so the source point coincides with it.

Properties Explorer

Start / End: Source and destination reference points.

Distance: Optionally enter displacement toward the destination numerically.

Tips

Use Copy if the original must remain in place.

Ribbon tab: Home	Group: Modify
 Command: Copy	Command ID: JDesign.Modify.Copy

Create repeated copies of selected objects.

Procedure

1. Select the objects and start Copy.

2. Specify the source base point.

3. Specify a destination base point. Continue clicking other destinations for additional copies.

4. Press Esc when all required copies are placed.

Properties Explorer

Base point: Reference point on the source.

Destination base point: Reference point for placing a copy.

Distance: Optionally enter distance in the destination direction.

Ribbon tab: Home	Group: Modify
 Command: 2D stretch	Command ID: JDesign.Modify.Stretch

Stretch endpoints or vertices of selected objects.

Procedure

1. Select the target 2D objects and start the command.
2. Specify opposite corners of a region enclosing the endpoints or vertices to move.
3. Specify a base point and destination.
4. Check which points move and which remain in the preview, then confirm.

Properties Explorer

First / Second region point: Define the rectangular selection region on screen.

Base point / Destination: Define stretch direction and distance.

Tips

This differs from moving entire objects. Exclude endpoints that must remain fixed from the region.

Ribbon tab: Home	Group: Modify
 Command: Rotate	Command ID: Object.Rotate

Rotate selected objects about an axis.

Procedure

1. Select the objects and run Rotate.

2. Specify the rotation reference point. For an arbitrary axis, select an axis line.

3. Enter the angle, check direction and Copy, then confirm.

Properties Explorer

Reference point / Axis: Define the rotation center.

Angle: Rotation amount. Start and end directions may also be used.

Source / Target plane: Align orientation using planes.

Copy: Choose whether to retain the originals.

Tips

This rotates existing objects. It differs from the modeling Revolve command, which creates a body from a profile.

Ribbon tab: Home	Group: Modify
 Command: Mirror	Command ID: Object.Reflect

Mirror selected objects about a specified axis or plane.

Procedure

1. Select the objects to mirror.
2. Specify the reference point, line or plane.
3. Choose whether to retain the originals, inspect the mirrored orientation and confirm.

Properties Explorer

Point / Line / Plane: Choose the appropriate mirror reference.

Copy: Retain the originals while creating mirrored copies.

Tips

For symmetric components, use snaps to define the symmetry axis or plane accurately.

Ribbon tab: Home	Group: Modify
 Command: Scale	Command ID: Object.Scale

Scale selected objects.

Procedure

1. Select the objects to enlarge or reduce.

2. Specify the fixed point.

3. Enter a scale factor, check whether originals should remain and confirm.

Properties Explorer

Fixed point: Center of scaling.

Factor: 2 doubles the size; 0.5 halves it.

Start / End: Use a reference length to determine the factor.

Distort / X Y Z factors: Apply different factors in different directions.

Copy: Choose whether to retain the originals.

Tips

Unlike print or layout scale, this changes the geometry's actual dimensions.

Ribbon tab: Home	Group: Modify
 Command: Delete	Command ID: Object.Delete

Delete selected objects.

Procedure

1. Select the objects to delete.

2. Check the highlighted targets and run the command.

Tips

This deletes data rather than hiding it. If deletion was accidental, use Undo immediately and inspect the result.

Ribbon tab: Home	Group: Modify
 Command: Explode	Command ID: SelectedObjects.Decompose

Explode supported selected objects such as blocks and polylines.

Procedure

1. Click or window-select the objects to explode.
2. Run the command.
3. Check that the constituent objects can be selected individually.

Tips

Only one level is exploded. Dynamic blocks become their current geometry, while nested block references remain.

Retain an editable copy of the original.

Ribbon tab: Home	Group: Modify
Command: Offset	Command ID: Aequidist

Create an offset curve.

Procedure

1. Select the reference curve.
2. Enter the distance and check inside versus outside.
3. Set corner handling and confirm.

Properties Explorer

Reference curve: Source curve to offset.

Distance: Signed distance from the source.

Minimum sharp-corner angle: 0° makes all corners sharp; 180° makes none sharp.

Tips

An offset too large for narrow regions can cause self-intersections or prevent creation.

Ribbon tab: Home	Group: Modify
 Command: Trim	Command ID: Trim

Trim objects at intersections.

Procedure

1. If needed, select a cutting reference under Trim to.

2. Under Trim, click the segment to remove.

3. Check the object shortened to the intersection.

Properties Explorer

Trim to: Cutting reference. Where omitted, nearby intersections are used.

Trim: Specify the object and segment to remove.

Tips

Pick the part to remove, not the part to retain.

Ribbon tab: Home	Group: Modify
 Command: Surface trim	Command ID: Constr.Face.SurfaceTrim

Trim a surface using an intersecting surface.

Procedure

1. Select the target surface.

2. Select a cutting surface that intersects it.

3. Choose the region to remove on one side of the intersection and execute.

Properties Explorer

Target surface: Surface being edited.

Cutting surface: Surface defining the boundary.

Region to delete: Select the side to remove.

Tips

The surfaces must actually intersect. Visual overlap on screen is insufficient.

Ribbon tab: Home	Group: Modify
 Command: Split	Command ID: Trim.Split

Split objects at intersections.

Procedure

1. Pick the target curve near the desired intersection.

2. Check the split position and execute.

3. Check that the resulting pieces can be selected separately.

Properties Explorer

Split: Specify the target object and location.

Tips

Unlike Trim, this retains both sides as separate objects.

Ribbon tab: Home	Group: Modify
Command: Join	Command ID: Join

Join selected curves into connected chains when endpoints, layer, color and line pattern match.

Procedure

1. Select the lines, arcs and other curves to connect.
2. Run the command.
3. Check that each endpoint-connected chain has been joined.

Tips

Eligible curves share layer, color and line pattern and have coincident endpoints. Resolve gaps or attribute differences first.

Ribbon tab: Home	Group: Modify
 Command: Remove overlapping objects	Command ID: RemoveOverlappingCurves

Consolidate duplicate or overlapping selected curves with matching layer, color, pattern and width.

Procedure

1. Select curves whose overlaps should be cleaned up.
2. Run the command.
3. Check that overlapping segments have been consolidated.

Tips

Curves must have the same layer, color, line pattern and width. The command does not indiscriminately remove lines with different attributes.

Ribbon tab: Home	Group: Modify
 Command: Extend	Command ID: Expand

Extend objects.

Procedure

1. Select the object near the endpoint to extend.

2. Specify the destination object and any additional distance.

3. Inspect the extension and confirm.

Properties Explorer

Object to extend: The curve whose endpoint changes.

Extend to: Destination object.

Distance: Additional extension; a negative value shortens the object.

Ribbon tab: Home	Group: Modify
 Command: Envelope	Command ID: JDesign.Modify.Envelope

Apply envelope cleanup to straight lines within a rectangle.

Procedure

1. Left-click the first corner of the processing region.

2. Left-click the opposite corner to process the enclosed lines.

3. Inspect the result.

Mouse Operations

Operation: Right-click the second point to delete within the region; Shift and left-click deletes intermediate portions.

Tips

The button used for the second point changes the operation. Use left-click for normal envelope processing.

Ribbon tab: Home	Group: Modify
 Command: Circular clip	Command ID: JDesign.ClipCopy.Circle

Clip content inside a circular boundary and copy it to another position.

Procedure

1. Select the source 2D objects if required.
2. Specify the circular region's center and radius.
3. Specify the destination and place the clipped content.

Inputs

Center / Radius: Circular clipping region.

Destination: Position for the clipped copy.

Tips

The original geometry remains. Text, dimensions and similar objects must lie entirely inside the boundary to be included.

Ribbon tab: Home	Group: Modify
 Command: Rectangular clip	Command ID: JDesign.ClipCopy.Rectangle

Clip content inside a rectangular boundary and copy it to another position.

Procedure

1. Select the source 2D objects if required.
2. Specify opposite corners of the rectangular region.
3. Specify the destination and place the copy.

Inputs

Opposite corners: Rectangular clipping region.

Destination: Position for the clipped copy.

Tips

Curves crossing the boundary are clipped to it. Text and dimensions are copied only if entirely inside.

Ribbon tab: Home	Group: Modify
 Command: Fillet	Command ID: Round.Off

Round objects at a shared corner.

Procedure

1. Select near the corner; specify a second object if needed.
2. Enter the radius and choose whether to trim the source objects.
3. Inspect the connecting arc and retained line extents, then confirm.

Properties Explorer

Position / Second object: The two objects forming the corner.

Radius: Fillet size.

Trim: Shorten source objects to the arc tangency points.

Tips

This rounds 2D curves. Use 3D fillet for solid edges.

Ribbon tab: Home	Group: Modify
 Command: Fillet all corners	Command ID: RoundMultiple

Round all shared corners.

Procedure

1. Select the polyline or path whose corners should be rounded.
2. Enter the radius.
3. Inspect the result at every corner and confirm.

Properties Explorer

Position: Target path or connected objects.

Radius: Radius applied at corners.

Tips

Short edges or tight corners may not accept the same radius. Inspect every corner.

Ribbon tab: Home	Group: Modify
 Command: Chamfer	Command ID: Cut.Off

Chamfer objects at a shared corner.

Procedure

1. Select the corner objects, specifying the second object if needed.
2. Set mode, length and angle.
3. Check source trimming and the chamfer line, then confirm.

Properties Explorer

Position / Second object: Objects to chamfer.

Mode: Choose how entered lengths are interpreted.

Length / Angle: Chamfer size and inclination.

Trim: Choose whether to trim the source objects.

Tips

This edits 2D geometry. Use 3D chamfer for solid edges.

Ribbon tab: Home	Group: Modify
Command: Chamfer all corners	Command ID: CutOffMultiple

Chamfer all shared corners.

Procedure

1. Select a path or other eligible object for all-corner chamfering.

2. Set mode, length and angle.

3. Inspect the whole shape and confirm.

Properties Explorer

Position: Chamfer target.

Mode / Length / Angle: Conditions applied at each corner.

Tips

To modify only certain corners, use the ordinary Chamfer command individually.

Ribbon tab: Home	Group: Modify
 Command: 3D fillet	Command ID: Fillet

Start 3D fillet.

Procedure

1. Select one or more edges to fillet.
2. Enter the radius. If selecting through faces or solids, check the full target set.
3. Verify the edges and radius, then execute.

Properties Explorer

Edges: Individual edges to fillet.

Face edges: Edges adjacent to the selected face.

Solid edges: Edges belonging to the shell or solid.

Radius: Fillet size.

Tips

Selecting all edges of a face may include unintended edges. Select edges individually for local work. A radius too large for nearby thicknesses or gaps can prevent creation.

Ribbon tab: Home	Group: Modify
 Command: 3D chamfer	Command ID: Chamfer

Create chamfers on solid edges.

Procedure

1. Specify the reference face and target edges.
2. Choose equal-distance or unequal-distance chamfer.
3. Enter distances, check the reference-face and opposite sides, then execute.

Properties Explorer

Mode: Switch between equal and unequal distances.

Reference face / Edge: Chamfer location.

Distance 1: Distance on the reference-face side.

Distance 2: Distance on the opposite side.

Tips

For unequal distances, the chosen reference face determines which side uses Distance 1 or Distance 2.

Ribbon tab: Home	Group: Modify
 Command: Rectangular array	Command ID: Copy.Matrix

Arrange multiple copies of selected objects in a grid.

Procedure

1. Select the objects to array.
2. Enter copy counts in each direction and horizontal/vertical spacing.
3. Set the grid angle, inspect the entire arrangement and execute.

Properties Explorer

Right / Left / Up / Down count: Copies in each direction from the original.

Horizontal / Vertical distance: Spacing between adjacent copies.

Angle: Orientation of the whole array.

Tips

Large counts create many objects. Check direction and spacing with small counts first.

Ribbon tab: Home	Group: Modify
 Command: Circular array	Command ID: Copy.Circular

Arrange multiple copies of selected objects around a circle.

Procedure

1. Select the objects to array.
2. Specify the array center and count.
3. Choose a full circle or angular step and enter the required angle.
4. Check whether copy orientations remain fixed or rotate, then execute.

Properties Explorer

Center: Array rotation center.

Count / Placement: Copy count and full-circle or angular-step mode.

Angle: Separation between copies in angular-step mode.

Orientation mode: Fixed or Rotate.

Tips

For hole patterns, specify the center accurately and ensure the final copy does not overlap the original.

Ribbon tab: Home	Group: Modify Model
 Command: Union	Command ID: Constr.Solid.Fuse

Join overlapping selected solids using a Boolean union.

Procedure

1. Select the solids to join.
2. Run the command.
3. Check that overlapping regions are resolved and the solids are joined.

Tips

This does not group lines or surfaces. Check that targets are solids with volume and occupy overlapping positions.

Ribbon tab: Home	Group: Modify Model
 Command: Subtract	Command ID: Constr.Solid.Difference

Subtract the second selected solid from the first.

Procedure

1. Select the solid to be cut first.
2. Select the cutting solid second.
3. Run Subtract and check that the overlapping region is removed.

Tips

Selection order matters. Reversing target and tool reverses the operation. If the solids do not intersect, the intended hole will not be cut.

Ribbon tab: Home	Group: Modify Model
 Command: Intersect	Command ID: Constr.Solid.Intersect

Replace selected solids with their shared volume.

Procedure

1. Select the solids whose common region should remain.
2. Run Intersect.
3. Check that only the region contained in both solids remains.

Tips

Unlike Union, this discards non-overlapping regions. Use solids sharing a nonzero volume.

Ribbon tab: Home	Group: Modify Model
 Command: Shell	Command ID: Constr.Solid.Shell

Remove specified opening faces and hollow the solid to a uniform thickness.

Procedure

1. Select one or more faces to remove as openings.
2. Enter the retained wall thickness.
3. Inspect the openings and internal shape, then execute.

Properties Explorer

Opening faces: Faces removed during shelling.

Thickness: Retained wall thickness.

Tips

Narrow areas or tight curvature may prevent creation of inner faces at the chosen thickness.

Location: Home	Group: Modify Model
Hole Basic Shape and Entrance	Dialog: Hole

Overview

Create holes at specified positions on faces. Open Home > Modify Model > Hole, the large button beside Shell. The CAD view remains usable while the dialog is open.

Procedure

1. On Hole shape, set type, diameter, end condition, bottom and entrance.

2. On Placement, choose a method and specify positions in the CAD view.

3. Check the green position circles and hole count.

4. Use Apply and continue to create holes and continue placing them. Create and close creates the holes and closes the dialog.

Hole Types

Plain hole: Enter diameter directly. Bolt clearance hole: Choose a nominal bolt size to set drilling and counterbore dimensions. Tap drill (thread information): Add thread metadata and simplified graphics. Reamed hole (tolerance information): Specify diameter and tolerance information.

End Condition and Bottom

Choose Through selected solid or Blind. Blind depth is the cylindrical length excluding the drill tip. Choose Drill point or Flat bottom; for a drill point, enter the tip angle.

Entrance Settings

Choose None, Counterbore, Countersink or Spotface. A counterbore uses drill diameter d', counterbore diameter D and depth H2. A countersink uses entrance diameter and angle. Registered bolt sizes range from M1.6 to M30.

Input Example

The registered M6 values are drill diameter 6.6 mm, counterbore diameter 11 mm and depth 6.5 mm. Custom dimensions allows direct entry. Check the values against the actual bolt and machining conditions.

Location: Home	Group: Modify Model
Hole Repeated Clicks and Array Placement	Dialog: Hole > Placement

Repeated Click Placement

1. Select Repeated clicks as the placement method.

2. Click successive hole centers in the CAD view. Each clicked face position and local normal determine that hole's position and direction.

3. Continue on different faces if required. Later clicks do not change directions of previously placed holes.

Circular Placement

1. Choose Circular placement and set count, PCD, start angle and distribution angle.

2. Click the array center on a reference planar face.

3. A 360-degree distribution spaces holes evenly around the full circle without duplicating the endpoint. A negative angle reverses direction.

Rectangular Placement

1. Choose Rectangular placement and set X/Y counts, spacings and array rotation.

2. Click the start point on a plane. Counts include the start position. Set one direction to a count of 1 for a single row.

Orientation and Targets

Zero degrees follows world X projected onto the reference face. If the face is normal to world X, world Y is used. Circular and rectangular arrays lie on the reference plane.

Check and Undo Placement

Check green circles and the list. Undo last removes the previous placement; Clear positions removes uncreated placements. The count is not fixed at six.

Holes Crossing Boundaries

The cutting tool is not clipped to the face outline, so edge-crossing and half holes are possible. Only the solid owning the selected face is cut. Hidden or locked solids are excluded.

Location: Home	Group: Modify Model
Hole Thread Information and Simplified Display	Dialog: Hole > Thread

Overview

Store thread designation, pitch and other information on a drilled hole, with simplified model and layout graphics. This does not create a solid helical thread.

Procedure

1. Choose Tap drill (thread information) and open Thread.
2. Enable thread information and simplified display, then choose designation and pitch P.
3. Set thread depth, start distance from the entrance, handedness and tolerance class.
4. Enable model and layout displays as required, then place and create the holes.

Dimensional Data

Basic dimensions for designation and pitch are read from ANASIZE.dat. If several pitches exist, the first registered value is the default. Set basic minor diameter D1 as drill diameter copies a basic dimension; it is not a recommended machining drill size.

Model Display

Show thread pattern in model adds simplified lines along the hole's inner face, excluding counterbore and countersink regions. Tolerance class and handedness are stored as thread information.

Layout Display

Show thread drafting lines in layout adds concentric diameter circles in axial views and thread-extent lines in side views. Obscured portions use hidden lines. Incomplete-thread display is configured on Thread depth.

Tips

Show thread to end of cylindrical portion displays the thread to the hole end. Disable it to specify thread depth independently, and check its relationship to drilled depth.

Location: Home	Group: Modify Model
Hole Automatic Thread-depth Calculation	Dialog: Hole > Thread Depth

Overview

Set suggested thread and drill depths, then override individual values as needed. D is nominal thread diameter and P is pitch. Automatic values are machining guides, not checks of strength or tool suitability.

Procedure

1. Select a blind hole on Hole shape, and set designation and pitch on Thread.
2. Disable thread display to the cylindrical end and select a depth basis on Thread depth.
3. Check thread depth, incomplete thread, bottom allowance, cylindrical drill depth and calculated results.
4. Edit required values and confirm with Apply and continue, or Apply changes while editing.

Automatic Defaults

Thread depth can be 1D, 1.5D or 2D, with 2D as default. Incomplete thread defaults to 2.5P and bottom allowance to 1P. Specify additional tool-tip length for the actual tool.

Drilled Depth

Automatic cylindrical depth equals thread start position + thread depth + incomplete thread + bottom allowance + additional tool-tip length. The start is the deeper of Start from entrance and the end of entrance machining. The conical drill tip is added beyond the cylindrical depth.

Calculation Example

With no entrance machining, start 0 mm, M6×1, 2D and additional tool-tip length 0 mm: thread depth is 12 mm, incomplete thread 2.5 mm, bottom allowance 1 mm and suggested cylindrical drill depth 15.5 mm.

Manual Overrides and Automatic Values

Typing a value makes only that item manual. Use Automatic: 2D, Automatic: 2.5P, Automatic: 1P or Calculate automatically to restore its automatic value. If a manually set drilled hole is too short, increase it to contain the thread range.

Create a circular pattern around a flange's central hole

1. In the Hole command, select a circular pattern and set the count, pattern diameter, and other values.

2. Click the planar face to machine. The location is not yet confirmed at this stage.

3. Use Center snap to set the center of the central hole as the pattern center. The machining face remains the planar face selected earlier.

4. Check the placement and machining direction, then apply. For a rectangular pattern, also select the planar face before specifying the reference position.

Clicking to redefine the center changes only the center location. Use Clear positions to select the planar face again, or Undo last to return one step.

Re-edit holes

Double-click the inner wall, step face, or bottom of a hole created with the Hole command to edit the holes created in that operation. Change the settings and choose Apply changes or Apply changes and close. Other machining that does not affect these holes is retained.

For supported geometry, holes can also be edited after moving the entrance or rear face along the hole axis, while retaining the new thickness. If a fillet or cut overlaps the opening and reconstruction is not possible, the original geometry is retained. This does not convert holes in STEP files without history into Hole-command features.

Location: Home	Group: Modify Model
 **Hole Double-click Editing**	Dialog: Hole Editing

Overview

Double-click a hole created by the Hole command to change dimensions and thread information. Holes created together in one operation are edited as a group.

Procedure

1. In normal selection, double-click the inner face, counterbore shoulder or bottom of a hole.

2. When the edit dialog opens, check highlighted holes and target count.

3. Change diameter, entrance, depth or thread parameters.

4. Apply changes keeps the editor open; Change and close applies changes and closes it.

Targets and Placement

Target faces and position circles stay fixed as the mouse moves. Editing preserves hole positions and directions while changing shape parameters.

Saving and Moving

Editing information is saved in JDC. Holes remain editable after moving or rotating the whole solid, and in copied solids. Editing a copy does not modify the original.

When Re-editing Is Unavailable

Later operations such as face moves or fillets may invalidate the recorded hole shape and prevent re-editing. Holes created by other methods, such as ordinary extrusion subtraction, are not supported.

Apply and Cancel

Close discards unapplied inputs. Cancel calculation interrupts processing. Failed operations are not applied and retain the input model. Use Undo to reverse committed changes.

Ribbon tab: Home	Group: Modify Model
 Command: Section curves	Command ID: PlaneIntersection

Intersect a solid, shell or face with a plane.

Procedure

1. Select the solid, shell or face whose section curves are required.

2. Specify the intersection plane.

3. Inspect the intersection curves and confirm.

Properties Explorer

Plane: Section position and orientation.

Tips

This creates intersection curves. Use Split solid by faces to divide the actual body.

Ribbon tab: Home	Group: Modify Model
 Command: Split solid by faces	Command ID: Constr.Solid.SplitByFace

Split a selected solid with a face or open shell, retaining solids on both sides.

Procedure

1. Specify the solid to split.
2. Specify the cutting face, or a face belonging to an open shell.
3. Check the cutting position and execute.

Properties Explorer

Solid: Body to split.

Cutting face: Surface defining the split boundary.

Tips

Both sides are retained. Inspect the result and delete the unwanted side if only one is needed.

Ribbon tab: Home	Group: Modify Model
 Command: Solid interference check	Command ID: Constr.Solid.InterferenceCheck

Start solid interference checking.

Procedure

1. Select multiple solids to check.
2. Run Interference check.
3. Inspect the results to find overlapping regions.

Tips

Touching and overlapping volumes are different conditions. Recheck after changing component positions.

Inspect slicing and interference results

For solid slicing, select the solid and cutting face. Confirm the cutting surface crosses the solid, then select each resulting part and check the split position.

For interference checking, select the participating solids. Original solids remain on their original layers; intersection results are created on the interference-check layer. Distinguish original parts from result geometry when repeating the operation.

When selecting filled symbols imported from DWG, also check layer locks and selection filters. Updating the software does not reconstruct geometry or editing information already lost.

Ribbon tab: Home	Group: Modify Model
 Command: Fillet	Command ID: Fillet

Start the Fillet command.

Procedure

1. Select one or more edges to fillet.
2. Enter the radius. If selecting through faces or solids, check the full target set.
3. Verify the edges and radius, then execute.

Properties Explorer

Edges: Individual edges to fillet.

Face edges: Edges adjacent to the selected face.

Solid edges: Edges belonging to the shell or solid.

Radius: Fillet size.

Tips

Selecting all edges of a face may include unintended edges. Select edges individually for local work. A radius too large for nearby thicknesses or gaps can prevent creation.

When slicing, chamfering, or filleting fails

Check that the cutting face crosses the solid, and inspect boundaries around holes and protrusions, intersecting faces, and very thin remaining walls. The surrounding geometry limits chamfer widths and fillet radii. A smaller value does not always succeed.

Inspect the result after slicing, hole-edge chamfering, filleting or protrusion editing. Opening a drawing does not automatically restore geometry damaged by an earlier operation or recover missing protrusions. Keep a copy saved before the operation and inspect the affected area.

Fillet multiple edges and inspect connections

For supported pipe/flange junctions, a fillet of the specified radius can terminate at the side boundary. Select the junction edge in the normal fillet command and enter the radius. Inspect the termination and retained holes/neighbouring parts, including a section view.

Select a fillet face with Change Fillet Size to alter its radius, or use Remove Fillet to restore the junction. Supported pipe branches, circular flanges and stepped pipe shoulders are included. Recoverable support faces and valid geometry are required; arbitrary radii and all imported shapes are not guaranteed. Inspect the committed result and use Undo/Redo as needed.

Select the boundary edges to round, specify the radius and confirm. Edges requiring the same radius can be selected together. Select the boundary between the two intended support faces rather than the face itself.

Inspect the resulting rounds, their end connections and connections to adjacent rounds, and check the intended holes and ribs remain intact. Available radius depends on local thickness, face spacing and interference between blends. A small radius is not guaranteed to succeed.

To remove rounds, select the fillet faces and inspect the result. Supported geometry is restored by extending the support faces to the original corner. If restoration is not geometrically possible, use Undo or the drawing saved before the operation.

On circular flanges, distinguish inner and outer circles and upper and lower edges. For an equal-distance chamfer at the top of a central hole, select that inner rim and enter the distance. Use a section view to confirm the selected boundary.

Ribbon tab: Home	Group: Modify Model
 Command: Chamfer	Command ID: Chamfer

Create chamfers on solid edges.

Procedure

1. Specify the reference face and target edges.
2. Choose equal-distance or unequal-distance chamfer.
3. Enter distances, check the reference-face and opposite sides, then execute.

Properties Explorer

Mode: Switch between equal and unequal distances.

Reference face / Edge: Chamfer location.

Distance 1: Distance on the reference-face side.

Distance 2: Distance on the opposite side.

Tips

For unequal distances, the chosen reference face determines which side uses Distance 1 or Distance 2.

Ribbon tab: Home	Group: Modify Model
 Command: Remove fillet	Command ID: Fillet.Remove

Remove fillet faces and restore the sharp corner.

Procedure

1. Select the fillet faces to remove.
2. Check which neighboring faces will be extended to restore the corner.
3. Execute and verify the restored corner.

Properties Explorer

Fillet faces to remove: Select the faces forming the round.

Tips

Select fillet faces, not edges. Later edits to adjacent faces may prevent restoration.

Location: Home	Group: Modify Model
Change fillet size	Command ID: Fillet.ChangeRadius

Overview

Select an existing 3D fillet and change its radius. This is separate from creating a new fillet.

Procedure

1. Choose Home > Modify Model > Fillet drop-down > Change fillet size.
2. Click the curved fillet face, not an edge.
3. Enter the new radius. The first selected fillet's current radius supplies the initial value.
4. Check highlighted faces and the value, then confirm with the check mark.

Select Multiple Fillets

Smoothly connected faces with the same radius are selected together. You can add separate fillets or other solids. Click a selected group again to remove it.

During Processing

Targets are highlighted during input; geometry is calculated when confirmed. Cancel from the progress display or press Esc during calculation. One Undo reverses the grouped changes.

Tips

Fillets can also be recognized after saving and reopening. Not every freeform or interfering shape is supported. If selection or processing fails, check targets and radius. Failure or cancellation preserves the original model.

Ribbon tab: Home	Group: Modify Model
 Command: Remove faces	Command ID: Face.Remove

Remove opening faces and restore surrounding faces where possible.

Procedure

1. Select the opening or other faces to remove.
2. Check the targets and execute.
3. Check that surrounding faces are restored and unwanted openings are gone.

Properties Explorer

Faces to remove: Faces removed from the shape.

Tips

This changes geometry rather than merely hiding faces. Shapes that cannot be healed from surrounding faces may fail.

Ribbon tab: Home	Group: Modify Model
 Command: Move faces	Command ID: Face.Move

Move selected faces to change a solid's shape.

Procedure

1. Select the faces of the hole, protrusion or other feature to move.

2. Specify the source base point.

3. Specify the destination, check connections to surrounding geometry and execute.

Properties Explorer

Faces to move: The feature whose position changes.

Base point / Destination: Movement direction and distance.

Tips

This relocates the selected feature rather than the whole model. Include all related faces.

Edit Faces Across Multiple Solids

Move faces, Offset faces and Taper faces can target faces on different solids together. The solids need not be joined with Union first.

Example of Aligning Heights

1. Choose an isometric or other clear view and run Move faces.

2. Click or window-select the top faces of several posts. Check that no side faces are included.

3. Set a corner of the column top as the base point and the corresponding corner of the beam top as the destination. Check the endpoint snap position.

4. After applying, check each column top and verify that the solids remain closed. One Undo reverses the entire operation.

Operation	Reference and movement	Selection checks
Move faces	Common direction from base point to destination	Distinguish moving faces from fixed faces
Offset faces	Outward normal of each face and distance	The same positive value moves top and bottom faces in opposite directions
Taper faces	Common fixed plane and angle	Fixed faces and boundaries; range in which valid geometry can be created

Applying the operation

Selected faces are highlighted during input; geometry is calculated when you apply. All targets are updated together only after every calculation succeeds. If any target fails, the original geometry is retained so that you can adjust the input.

Difference from a dynamic block

Ordinary face editing changes the current geometry. To link a Length input to face movement in a reusable component, register a 3D face action as described in Part 3. Save the registered parameters and actions in JDC format.

Move faces while retaining protrusions and holes

Use Move faces to select the faces belonging to the moving protrusion and hole together, specify the destination and confirm. With supported planar arrangements, the base plate extends while protrusion thickness, hole diameter and the stationary fillet are retained.

This selection differs from moving only the root face. Inspect the hole, sides and root after moving, rotating or copying. A move that intrudes into an existing round or causes self-intersection may not be valid.

Ribbon tab: Home	Group: Modify Model
 Command: Rotate faces	Command ID: Face.Rotate

Rotate selected hole or protrusion faces within the same plane.

Procedure

1. Select the hole or protrusion faces.
2. Specify the rotation center.
3. Specify the angle, check placement and surrounding geometry, then execute.

Properties Explorer

Faces to rotate: Hole or protrusion to move.

Rotation center: Center about which the faces rotate.

Angle: Rotation amount.

Tips

Use this to rotate the placement of holes or protrusions on one face. Use Modify > Rotate to change the entire model's orientation.

Ribbon tab: Home	Group: Modify Model
 Command: Copy faces	Command ID: Face.Copy

Copy selected holes or protrusions.

Procedure

1. Select their faces.
2. Specify the source base point and destination.
3. To place on another face, specify the destination face.
4. Inspect the new shape and execute.

Properties Explorer

Faces to copy: Hole or protrusion to duplicate.

Base point / Destination: Source and target reference positions.

Destination face: Target when copying to another face.

Ribbon tab: Home	Group: Modify Model
 Command: Offset faces	Command ID: Face.Offset

Offset multiple selected faces along their individual outward normals.

Procedure

1. Select one or more faces to offset.
2. Enter the offset distance.
3. Check inward versus outward change and execute.

Properties Explorer

Faces to offset: Faces whose positions change.

Offset distance: Displacement relative to each outward normal.

Tips

Each face moves along its own normal. This differs from translating all faces in one common direction.

Ribbon tab: Home	Group: Modify Model
 Command: Taper faces	Command ID: Face.Taper

Apply a common taper angle to multiple selected planar faces.

Procedure

1. Select faces to taper.

2. Select the reference boundary edge at the neutral plane.

3. Enter the angle or use the gizmo, check the tilt side and execute.

Properties Explorer

Faces to taper: Supported planar or cylindrical target faces.

Reference boundary edge: A line or arc defining the neutral-side reference.

Taper angle: Positive values incline inward away from the neutral plane.

Tips

Choosing a different reference boundary also changes which side remains fixed.

Ribbon tab: Home	Group: Modify Model
 Command: Project curves onto faces	Command ID: Curve.ProjectToFace

Project closed sketch curves onto a curved face and split it along the projected edges.

Procedure

1. Select closed sketch curves. Multiple curves must connect at their endpoints to form a closed loop.

2. Select the target face.

3. Set projection direction and intersection-selection method, then execute.

Properties Explorer

Curves to project: Outline transferred to the face.

Target face: Face split by projected edges.

Projection direction: Projection vector.

Intersection selection: Rule used when a projection line meets the face more than once.

Tips

Projection splits the face; it does not by itself create a hole.

Project curves onto a face

Select the closed curve, target face and projection direction, then confirm. The direction must reach the target face from the curve. A direction with no intersection cannot split the face.

Models containing complex fillet faces may require time for processing and redraw. After completion, inspect the projected boundary, face split and closed solid.

Ribbon tab: Home	Group: Modify Model
 Command: Create section plane	Command ID: SectionPlane.Create

Create a section plane stored in the model (SECTIONPLANE).

Procedure

1. Define the plane using a face, three points or the current workplane.
2. Check display width and height.
3. Confirm to create a section-plane object in the drawing.

Properties Explorer

Section plane: Cutting position and direction.

Display width / Height: Size of the plane's visible frame.

Tips

These dimensions control the display frame, not the extent of the section calculation.

Ribbon tab: Home	Group: Modify Model
 Command: Toggle live section	Command ID: SectionPlane.Live

Toggle live section display using a selected section plane (LIVESECTION).

Procedure

1. Select the saved section plane's frame or arrow.
2. Toggle live section to inspect the model interior.
3. Toggle the same plane again to restore the original view.

Properties Explorer

Select section plane: Plane used for live display.

Tips

This clips the display; it does not cut or delete actual solid geometry.

Ribbon tab: Home	Group: Modify Model
 Command: Section curves to XY plane	Command ID: SectionPlane.Generate2D

Create exact solid intersection curves as a 2D block on the XY plane (SECTIONPLANETOBLOCK).

Procedure

1. Select the section plane to convert to 2D.
2. Specify the block placement position on the XY plane.
3. Confirm and inspect the generated section curves.

Properties Explorer

Select section plane: Reference plane for intersections.

2D placement point: Base point of the new block.

Tips

The result is a 2D block of solid-plane intersection curves, not an image of the live section.

Ribbon tab: Home	Group: Modify Model
 Command: Section settings	Command ID: SectionPlane.Settings

Set live display, cutting direction and display extents (SECTIONPLANESETTINGS).

Procedure

1. Select the target section plane's frame or arrow.
2. Set live section, cutting direction, display width and height.
3. Check the displayed side and confirm.

Properties Explorer

Live section: Enable or disable display clipping.

Reverse cutting direction: Switch the side retained in the display.

Display width / Height: Section-plane frame size.

Tips

Display-extent settings are separate from actual solid cutting.

Ribbon tab: Home	Group: Annotation
 Command: Automatic dimension	Command ID: Constr.Automatic.Dimension

Choose a dimension type automatically from the input method and selected geometry.

Procedure

1. Select a line, circle or arc, or specify the first dimension point.
2. For a two-point length, specify the second point; for an angle, select the second line.
3. Move the mouse, check dimension type and position, then place it.

Properties Explorer

Object / First point: Choose object selection or point input.

Second line / Position / Second point: Supply the next input appropriate to the target.

Dimension-line position: Where to place the dimension.

Extension-line angle: Normally 90° to the dimension line.

Tips

If the inferred type is not what you need, use a dedicated Horizontal, Vertical, Radius or other dimension command.

Ribbon tab: Home	Group: Annotation
 Command: Text	Command ID: Constr.Text.1Point

Text: Specify a reference point.

Procedure

1. Set text content, font and size.
2. Specify the text reference point.
3. Check direction and position, then confirm.

Properties Explorer

Content: Text to place in the drawing.

Font / Text height: Typeface and size.

Reference point: Text placement position.

Tips

To display measured values as dimensions, use dimension commands rather than manually entered text.

Ribbon tab: Home	Group: Annotation
 Command: Text by two points	Command ID: Constr.Text.2Points

Text: Specify start and end points.

Procedure

1. Set text content and formatting.
2. Specify the text reference point.
3. Specify the second baseline point, check text direction and appearance, then confirm.

Properties Explorer

Reference point: Text placement reference.

Second point: Defines baseline direction.

Content / Font: Text string and typeface.

Tips

Use this for notes aligned by two points, such as text along an inclined line.

Ribbon tab: Home	Group: Annotation
 Command: Convert text outlines to paths	Command ID: SelectedObjects.TextOutlinePath

Convert text outlines to paths.

Procedure

1. Select the text to convert.
2. Run the command.
3. Check that outer outlines and internal holes have been extracted as paths.

Tips

Converted outlines are no longer edited like text strings. Retain a copy of the original text if required.

Ribbon tab: Home	Group: Annotation
 Command: Multiline text	Command ID: Constr.Text.MText

Create multiline text inside a text area.

Procedure

1. Specify the first and opposite corners of the text area.
2. Enter the text in the multiline-text editor.
3. Check wrapping and area size, then confirm placement.

Properties Explorer

First / Opposite point: Text placement area.

Content / Formatting: Set these in the editor.

Tips

Long notes are easier to edit as multiline text than as separate stacked single-line objects.

Ribbon tab: Home	Group: Annotation
 Command: Horizontal	Command ID: Constr.Horizontal.Dimension

Create horizontal length dimensions.

Procedure

1. Specify the two measurement points.
2. Check that the dimension is horizontal and specify its line position.
3. For more dimensions, choose a mode and specify the next point.

Properties Explorer

Points: Origins of extension lines.

Dimension-line position: Offset placement away from the geometry.

Mode: Simple, Continuous, Chain or Baseline.

Extension-line angle: Normally 90°.

Baseline spacing: Distance between successive baseline dimensions.

Tips

Use Center snap for both ends of a center-to-center measurement. In a layout, check the target view and displayed value.

Ribbon tab: Home	Group: Annotation
 Command: Vertical	Command ID: Constr.Vertical.Dimension

Create vertical length dimensions.

Procedure

1. Specify the two measurement points.
2. Check that the dimension is vertical and specify its line position.
3. For more dimensions, choose a mode and specify the next point.

Properties Explorer

Points: Origins of extension lines.

Dimension-line position: Offset placement away from the geometry.

Mode: Simple, Continuous, Chain or Baseline.

Extension-line angle: Normally 90°.

Baseline spacing: Distance between successive baseline dimensions.

Tips

Use Center snap for both ends of a center-to-center measurement. In a layout, check the target view and displayed value.

Ribbon tab: Home	Group: Annotation
 Command: Length	Command ID: Constr.Sloping.Dimension

Create aligned length dimensions.

Procedure

1. Specify the two measurement points.
2. Check the measurement direction and specify the dimension-line position.
3. For more dimensions, choose a mode and specify the next point.

Properties Explorer

Points: Origins of extension lines.

Dimension-line position: Offset placement away from the geometry.

Mode: Simple, Continuous, Chain or Baseline.

Extension-line angle: Normally 90°.

Baseline spacing: Distance between successive baseline dimensions.

Direction: Direction along which length is measured.

Tips

Use Center snap for both ends of a center-to-center measurement. In a layout, check the target view and displayed value.

Ribbon tab: Home	Group: Annotation
 Command: Between objects	Command ID: Constr.Distance.Dimension

Create a distance dimension between two objects.

Procedure

1. Select the first measurement object.
2. Select the second object.
3. Check the measurement locations and line, then place the dimension.

Properties Explorer

First / Second object: Measurement targets.

Dimension-line position: Placement location.

Extension-line angle: Angle relative to the dimension line.

Tips

Pick locations may change the candidate distance. Check that the intended features are measured.

Ribbon tab: Home	Group: Annotation
 Command: Radius	Command ID: Constr.Radius.Dimension

Create a radius dimension.

Procedure

1. Select the circle or arc to dimension.
2. Move the mouse to choose the radius text position.
3. Check the R-prefixed value and leader direction, then place it.

Properties Explorer

Circle or arc: Radius target. Select the circumference or arc, not its center.

Dimension-line position: Text and leader placement.

Tips

For a large arc whose center is off screen, select the visible arc. Check the object and scale instead of manually overriding its value.

Ribbon tab: Home	Group: Annotation
 Command: Diameter	Command ID: Constr.Diameter.Dimension

Create a diameter dimension.

Procedure

1. Select the circle or arc to dimension.
2. Specify the dimension-line position.
3. Check the diameter symbol and value, then place it.

Properties Explorer

Circle or arc: Diameter target.

Dimension-line position: Diameter-dimension placement.

Tips

This displays diameter, not radius. For holes, select the correct circumference and distinguish inner from outer diameter.

Ribbon tab: Home	Group: Annotation
 Command: Angle between objects	Command ID: Constr.Angle.Dimension.Objects

Create an angular dimension from lines or arcs.

Procedure

1. Select the first angle reference object.
2. Select the second line.
3. Place the dimension on the side containing the required angle.

Properties Explorer

Arc or line: First object.

Line: Second object.

Dimension-line position: Determines the side and placement of the angle.

Tips

Interior and exterior angles differ. Check that the dimension arc spans the intended region.

Ribbon tab: Home	Group: Annotation
 Command: Three-point angle	Command ID: Constr.Angle.Dimension.3p

Create an angular dimension from three points.

Procedure

1. Specify the angle vertex.
2. Specify a point defining the first side.
3. Specify a point defining the second side.
4. Specify the dimension-line position.

Properties Explorer

Center: Angle vertex.

First / Second angle point: Define the two side directions.

Dimension-line position: Angular-dimension placement.

Tips

Specify points in this order: vertex, point on the first side, point on the second side.

Ribbon tab: Home	Group: Annotation
 Command: Four-point angle	Command ID: Constr.Angle.Dimension.4p

Create an angular dimension from four points.

Procedure

1. Specify two points on the first side in order.
2. Specify two points on the second side in order.
3. Check dimension placement and angular range, then place it.

Properties Explorer

First / Second point: Direction of the first side.

Third / Fourth point: Direction of the second side.

Dimension-line position: Angle-label placement.

Tips

Use this when the intersection is difficult to pick directly. Each pair of side points must be distinct.

Ribbon tab: Home	Group: Annotation
 Command: Chamfer dimension	Command ID: Constr.Chamfer.Dimension

Create a C dimension for a 45-degree equal-distance chamfer.

Procedure

1. Select the 45-degree equal-distance chamfer.
2. Specify the text position.
3. Check that it displays as a C dimension.

Properties Explorer

Chamfer edge: Chamfer to dimension.

Dimension position: Text placement.

Tips

This is for 45-degree equal-distance chamfers. Use length and angle dimensions for other angles or unequal distances.

Ribbon tab: Home	Group: Annotation
 Command: Leader	Command ID: Constr.Labeling.Dimension

Create a leader.

Procedure

1. Specify the leader target point.

2. Specify successive bend points toward the text location.

3. Enter the note and confirm where line and text do not overlap.

Properties Explorer

Target point: Location being annotated.

Leader points: Required bend points.

Text: Annotation content.

Mode: Change the input method if needed.

Ribbon tab: Home	Group: Annotation
Command: Surface Label	Command ID: MenuId.Constr.SurfaceLabel

Display text or images as a sticker on faces without changing solid geometry or volume.

Procedure

1. Choose Home > Annotation > Leader dropdown > Surface Label, then click the target face.
2. Select Add and choose Text or Image. Enter text or select an image file.
3. Adjust width, height, rotation and center, inspect the preview, then confirm with the check mark.

Edit or remove

Pick the face and choose the existing item from Target Label. Set Operation to Edit or Remove, then confirm. Picking a face alone does not finish the command. Undo restores the previous state.

Settings

Font Name initially uses the current text font and offers a font list. Width and height use millimetres; rotation uses degrees.

Planes, cylinders, spheres and freeform surfaces are supported. Labels span connected patches of the same surface and periodic seams, and clip at boundaries and holes. Curvature can distort the label near its edges.

Labels are embedded in JDC, follow move/copy operations and appear in printing and PNG output. STEP and STL do not contain label appearance.

Ribbon tab: Home	Group: Annotation
 Command: Coordinate leader	Command ID: Constr.Point.Dimension

Create a coordinate annotation for a point.

Procedure

1. Specify the point or object whose coordinates are required.
2. Specify the coordinate-text position.
3. Check the displayed coordinates and confirm.

Properties Explorer

Target point: Position from which coordinates are read.

Dimension-line position: Coordinate-note placement.

Mode: Change the point-selection method if needed.

Tips

Text placement and measurement location are separate. Snap accurately to the point being measured.

Ribbon tab: Home	Group: Annotation
 Command: Balloon	Command ID: Constr.Labeling.Balloon

Start Balloon.

Procedure

1. Check item number, prefix and suffix in symbol settings.
2. Specify the arrow position.
3. Specify the balloon-circle center to place it.

Properties Explorer

Arrow position: Target location on the component.

Circle center: Position of the numbered circle.

Symbol / Prefix / Suffix: Displayed content.

Numbering: Numeric, Alphabetic or None.

Tips

For parts-list aggregation, keep item numbers and component attributes consistent.

Ribbon tab: Home	Group: Annotation
 Command: Marker	Command ID: Constr.Labeling.Marker

Start Marker.

Procedure

1. Set content and numbering in symbol settings.
2. Specify the marker-circle center.
3. Check number and placement, then confirm.

Properties Explorer

Circle center: Marker placement position.

Symbol / Prefix / Suffix: Displayed content.

Numbering: How successive identifiers advance.

Tips

Unlike a balloon, a marker is placed without specifying an arrow point.

Ribbon tab: Home	Group: Annotation
 Command: Geometric tolerance	Command ID: Constr.GeometricTolerance

Create a geometric-tolerance feature control frame.

Procedure

1. In the dialog, set the symbol, tolerance and required datums.

2. Confirm the settings.

3. Specify a drawing position for the frame.

Dialog and Placement Inputs

Tolerance symbol / Value: Specify the drawing's tolerance conditions.

Datums: Reference features.

Placement point: Position of the frame.

Tips

Enter tolerances according to design requirements. The command does not infer required tolerances from the model.

Ribbon tab: Home	Group: Annotation
 Command: Datum	Command ID: Constr.DatumIdentifier

Create a datum identification symbol.

Procedure

1. Place the target point on the reference face or location.
2. Specify the symbol position.
3. Check the identifier and referenced feature.

Properties Explorer

Target point: Location identifying the reference.

Symbol position: Placement of the datum frame.

Tips

Check automatically assigned letters too, and ensure they match references in feature control frames.

Ribbon tab: Home	Group: Annotation
 Command: Surface finish symbol	Command ID: Constr.SurfaceFinish

Start Surface finish symbol.

Procedure

1. Set the required conditions in the surface-finish dialog.
2. Specify target and bend points if a leader is used.
3. Specify the symbol position to place it.

Dialog and Placement Inputs

Symbol settings: Enter finish conditions and displayed content.

Target / Bend points: Leader positions.

Symbol position: Surface-finish symbol placement.

Tips

This does not measure surface roughness from model geometry. Enter the required specification.

Ribbon tab: Home	Group: Annotation
Command: Parts list	Command ID: Constr.Labeling.PartsList

Start Parts list.

Procedure

1. Prepare balloon or marker identifiers and component attributes.
2. Open the dialog and check aggregation results and required columns.
3. Edit names, quantities and other entries, and set the placement direction.
4. Confirm and specify the table's drawing position.

Parts List Dialog

Item / Part number / Name / Quantity: Inspect and edit table content.

Placement direction: Add downward from the top or upward from the bottom.

Export CSV: Export the table being edited as CSV.

Tips

Select an existing table and run the command again to edit it. Check regenerated totals after changing parts or identifiers.

Ribbon tab: Insert	Group: Insert
 Command: Insert block	Command ID: Insert

Insert a block or drawing.

Procedure

1. Choose Block or File as the insertion source.
2. Specify the block name or JDC file.
3. Set insertion point, scale and rotation, then place it. Enter attribute values if prompted.

Properties Explorer

Source / Block / File: Choose the data to insert.

Insertion point: Position of the block base point.

X / Y / Z scale: Scale factor in each direction.

Rotation angle: Insertion orientation. Explode on insertion: Initially No; subsequent operations retain the previous choice.

Tips

Scale 1 preserves physical size. Registered blocks use the definition base point; files use the drawing insertion base. Yes explodes only the outer level. See Part 3 for details.

Ribbon tab: Insert	Group: Insert
 Command: Insert CAD geometry	Command ID: Import

Add geometry from JDC, DWG, DXF or STEP files.

Procedure

1. Select the CAD file to insert.
2. Inspect imported geometry and specify an insertion position if prompted.
3. Check its position and size in the current drawing.

Tips

Insertion uses the drawing origin and retains original block references and supported dynamic information. Use Insert block to treat an entire drawing as one block.

Ribbon tab: Insert	Group: Insert
 Command: Insert SVG geometry	Command ID: Import.SVG

Add SVG geometry.

Procedure

1. Select the SVG file.
2. Inspect the imported objects and specify placement as required.
3. Check line and curve positions, sizes and closed outlines.

Tips

Some SVG representations may not convert identically to CAD lines and fills. Check closed geometry before using outlines for machining.

Ribbon tab: Insert	Group: Insert
 Command: Block	Command ID: BlockManager

Preview and insert JDC blocks from a folder library.

Procedure

1. Select a library tab and folder.
2. Choose a JDC block from the list and inspect its preview.
3. Check scale, angle and Explode on insertion, then insert.
4. Specify the drawing placement position.

Block Manager

Library tab / Folder: Location of the block library.

Preview: Inspect the shape before insertion.

Scale / Angle: Placement size and orientation.

Explode on insertion: Retain grouping or insert constituent objects.

Tips

Deleting a library tab removes its registration, not the files in its folder.

Ribbon tab: Insert	Group: Block Definition
 Command: Attribute definition	Command ID: Constr.Block.AttributeDefinition

Create an attribute to include in a block.

Procedure

1. Enter tag, prompt and default value.
2. Set text height, font and display options.
3. Specify the insertion point to place the attribute definition.
4. Select the component geometry together with its attribute definitions and create the block.

Properties Explorer

Tag: Attribute identifier.

Prompt / Default: Insertion prompt and initial value.

Insertion point / Text height / Font: Attribute-text placement and formatting.

Invisible / Constant and other options: Display and input behavior.

Tips

Including attribute definitions lets you enter numbers, names and other values for each inserted component.

Ribbon tab: Insert	Group: Block Definition
 Command: Create block	Command ID: Block

Create a named block definition from selected objects and replace them with a block reference.

Procedure

1. Select geometry and any attribute definitions to include.
2. Enter the block name.
3. Specify the insertion base point and confirm.

Properties Explorer

Name: Block name within the drawing.

Base point: Reference position for insertion and placement.

Component attributes: Set these if needed.

Tips

The selection is replaced by a block reference, suitable for managing reusable components.

Ribbon tab: Insert	Group: Block Definition
 Command: Export block	Command ID: WBlock

Save selected geometry as JDC while retaining coordinates and storing the specified drawing insertion base.

Procedure

1. Select the geometry to export.

2. Specify the block base point.

3. Choose a folder and file name to write the JDC.

Properties Explorer

Base point: Insertion base stored in the exported drawing. Drawings without one use the origin.

Component attributes: Set these if needed.

Tips

Use a mounting point or component center as the base for easier alignment when inserting elsewhere.

Ribbon tab: Insert	Group: Export
 Command: Export CAD geometry	Command ID: Export.Selection

Export selected objects.

Procedure

1. Select the CAD objects to export.
2. Specify format, folder and file name.
3. Export and, if needed, inspect the geometry in the receiving CAD application.

Tips

Available formats depend on format support and license. Choose one that preserves the solids, curves, colors and other information required.

Ribbon tab: Insert	Group: Export
 Command: Export SVG geometry	Command ID: Export.SVG

Export selected 2D objects as SVG.

Procedure

1. Select the 2D objects to export.

2. Specify folder and file name, then export.

3. Open the SVG and check extents and line appearance.

Tips

SVG is for 2D exchange; it does not retain 3D solids as such.

Ribbon tab: Insert	Group: Export
 Command: Export layout as 2D	Command ID: LayoutView.Export2D

Export the current layout as a cleaned 2D drawing in JDC, DWG, DXF or JWW.

Procedure

1. Activate the target layout.

2. Run the command and specify the format and destination.

3. Check view and dimension placement in the exported 2D drawing.

Tips

The layout is converted to 2D entities. Also save the original JDC to retain editable model associations.

Ribbon tab: Insert	Group: Reference
 Command: Attach	Command ID: ExternalReference.Attach

Load an external reference drawing.

Procedure

1. Select the reference file.
2. Check reference method and path settings.
3. Specify the insertion position.

Tips

External references depend on their source files. If a source is moved or renamed, check its path in Reference manager.

Ribbon tab: Manage/Measure	Group: Manage
 Command: Purge	Command ID: Purge

Remove unused definitions, layers and other named entries from the drawing.

Procedure

1. Open Purge.
2. Choose unused block definitions, layers or other entries to remove.
3. Check the targets and delete them.

Tips

This does not directly delete geometry. Unused definitions reserved for later work may also appear, so check what should be retained.

Ribbon tab: Manage/Measure	Group: Manage
 Command: Reference manager	Command ID: ExternalReference.Manager

Manage external references currently inserted in the drawing.

Procedure

1. Open Reference manager and check reference names and status.
2. Reload to reflect changes in a source file.
3. Correct paths for missing references and detach unwanted ones.

Reference Manager Operations

Load / Unload: Toggle reference display while retaining placement.

Reload: Apply source-file changes.

Change path: Update the reference file location.

Detach: Remove reference placements and definition.

Tips

Unlike Unload, Detach removes placements too. Check the effect on every insertion of the same reference.

Ribbon tab: Manage/Measure	Group: Measure
 Command: Distance between two points	Command ID: Measure.Calculate.Distance3D

Measure the distance between two points in 3D.

Procedure

1. Specify the first measurement point.
2. Specify the second point and inspect the result.
3. To retain the result in the drawing, specify a text position and confirm.

Properties Explorer

First / Second point: Measurement positions.

Result: Distance in 3D space.

Text position: Placement of the measurement note.

Tips

This is not a screen-projected distance. Different elevations include the Z difference in the calculation.

Ribbon tab: Manage/Measure	Group: Measure
 Command: UCS angle	Command ID: Measure.Calculate.UcsAngle

Project the direction between two points onto the current UCS and calculate its angle from UCS X over 0–360°. Place the result as text.

Procedure

1. Set the required reference UCS.
2. Specify the two points defining direction.
3. Inspect the result and specify a text position if required.

Properties Explorer

First / Second point: Direction runs from the first to the second.

Result: Angle of the direction projected onto the UCS.

Text position: Result placement.

Tips

Angles range from 0 to 360° relative to UCS X. Reversing point order reverses direction.

Ribbon tab: Manage/Measure	Group: Measure
 Command: Face angle	Command ID: Measure.Calculate.FaceAngle

Calculate the inclination between a selected face and the current UCS plane, and place the result as text.

Procedure

1. Set the reference UCS.
2. Select the face to measure.
3. Inspect the result and place result text if needed.

Properties Explorer

Face: Measurement target.

Result: Inclination of the face to the UCS plane.

Text position: Where to retain the result in the drawing.

Tips

This is a 0–90° angle relative to the current UCS plane, not a dihedral-angle measurement between two selected faces.

Ribbon tab: Manage/Measure	Group: Measure
 Command: Closed-geometry perimeter	Command ID: Measure.Calculate.Perimeter

Project a closed curve onto the current UCS, calculate its perimeter and place the result as text.

Procedure

1. Set the reference UCS.
2. Click a circle, closed path, or contour made from endpoint-connected lines and arcs.
3. Inspect the perimeter and optionally specify a text position to confirm. Check that the full-contour highlight matches the intended boundary.

Properties Explorer

Closed geometry: Closed curve to measure.

Result: Length of the outline projected onto the UCS.

Text position: Result placement.

Tips

Connected lines and arcs can be measured without joining. Gaps, branches and hidden segments prevent recognition.

Esc clears the preview. Check the UCS projection and reference axes before calculating.

Ribbon tab: Manage/Measure	Group: Measure
 Command: Area and section properties	Command ID: Measure.Calculate.SectionProperties

Calculate area, centroid and second moments of area about X and Y for closed geometry, and place result text.

Procedure

1. Set the section's reference UCS.
2. Click a circle, closed path, or contour made from endpoint-connected lines and arcs.
3. Inspect area, centroid and second moments of area. Check that the full-contour highlight matches the intended boundary.
4. Place result text if needed.

Properties Explorer

Closed geometry: A coplanar closed outline.

Result: Area, centroid and X/Y second moments of area.

Text position: Placement of the calculation note.

Tips

Connected lines and arcs can be measured without joining. Gaps, branches and hidden segments prevent recognition.

Esc clears the preview. Check the UCS projection and reference axes before calculating.

Ribbon tab: Manage/Measure	Group: Measure
 Command: Distance between faces	Command ID: Measure.Calculate.FaceDistance

Calculate the shortest distance between two faces and create a line joining the closest points. The current method is approximate, with a 0.01 mm accuracy setting.

Procedure

1. Select the first face.
2. Select the second and inspect the distance and connecting line.
3. Confirm to create the line between the closest points.

Properties Explorer

First / Second face: Distance-measurement targets.

Result: Distance between faces.

Tips

The calculation uses triangulated surfaces. The 0.01 mm setting is a computational tolerance, not a guarantee that every curved-surface result is within 0.01 mm of the exact answer.

Ribbon tab: Manage/Measure	Group: Measure
 Command: Solid volume and centroid	Command ID: Measure.Calculate.SolidMassProperties

Calculate total volume and centroid coordinates of one or more solids and create a centroid point. Mass and density are not included.

Procedure

1. Select one or more solids.
2. Inspect total volume and centroid coordinates.
3. Confirm to create a point at the centroid.

Properties Explorer

Solids: Closed bodies to calculate.

Result: Total volume and volume-weighted centroid coordinates.

Tips

This is not a mass calculation using material densities. Overlapping solids are summed individually; union them first if overlap must be counted only once.

Location: Layout	Group: Layout
Layout Display and View Management	Command ID: LayoutView / LayoutPatch

Where to Start

In Properties Explorer > Views, right-click a layout name or one of its child views. Layout-name commands and individual-view commands act on different targets.

Open a Layout

1. Choose Display at the top of the layout-name menu to switch to that sheet.

2. Rename and Delete in the same group act on that layout. Check its name before deleting.

View Visibility and Locking

Turning off Show view excludes it from display, printing and 2D export. It remains in the tree and can be restored there. Show all views restores them together.

Protect Placement

Lock view fixes sheet position, frame, scale and projection direction. Snapping to geometry and editing sheet dimensions and text remain possible, as do sheet zooming and panning.

New Layout View

New layout view adds an ordinary view. Use VIEWBASE and VIEWPROJ for related orthographic views.

Tips

Independent text and lines drawn directly on the sheet remain when a view is hidden. Lock internal movement controls the frame/content movement relationship and differs from Lock view.

Location: Layout	Group: Layout
Layout VIEWBASE Base View	Command ID: ViewBase

Overview

Place a model on paper as a base projection for orthographic views. This does not change the model's dimensions or orientation.

Procedure

1. Prepare the model and display the target layout.
2. Choose VIEWBASE (Base view) from the layout context menu.
3. Set direction, scale, display mode and projection method.
4. Move to the intended sheet position and click to place it.

Main Inputs

Choose front, top, right or another direction. Scale is relative to actual size: 1 means 1:1, 0.1 means 1:10, and 2 means 2:1. Display mode controls hidden lines and appearance; first-angle or third-angle projection controls view arrangement.

After Placement

Select a view to inspect scale and other properties. Use Add projected view (VIEWPROJ) for related directions. Paper size and view scale are set separately.

Tips

Views reference the source model. Text and dimensions drawn directly on the layout are sheet entities. Lock the view after arranging it to prevent accidental movement.

Location: Layout	Group: Layout
Layout Add Projected View	Command ID: ViewProject

Overview

Add another direction related in position and projection to a base view.

Procedure

1. Select one reference projection in the layout.
2. Right-click and choose Add projected view (VIEWPROJ).
3. Move above, below, left, right or diagonally from the reference. Inspect the corresponding projection candidate.
4. Click the required position. Repeat to add other views.

Inherited Settings

The new view uses the reference's model, scale, display mode and projection method. Some inherited values cannot be changed independently during insertion.

Check Projection Method

Which face appears on each side depends on first-angle or third-angle projection. Inspect the preview before placing it.

Tips

Choose a supported parallel projection as the reference. Perspective and section views are excluded. Use an auxiliary view to look normal to a particular face.

Location: Layout	Group: Layout
Layout Auxiliary View Normal to a Face	Command ID: ViewAuxiliary

Overview

Add a view looking straight at an inclined planar face, making holes and outlines on it easier to inspect.

Procedure

1. Select a VIEWBASE or VIEWPROJ view.
2. Right-click and choose Auxiliary view normal to face.
3. Click a planar face in the view. If seen edge-on, click its edge.
4. Check Viewing side and Placement method, then click the sheet position.

Inputs

Reference plane/edge identifies the candidate face. Viewing side switches front and back. Placement method offers Align with projection direction or Free placement.

Result

The new view inherits model, scale, visible layers and visible/hidden-line settings. It projects the entire model from that direction, not only the selected face.

Tips

Only planar faces are supported; curved faces, perspective views and section views are excluded. Direction is set at creation, so check it after changing the reference face's inclination. Esc cancels.

Location: Layout	Group: Layout
Layout Add Circular Detail View	Command ID: ViewDetail

Overview

Extract a circular area from a source view and place an enlarged detail.

Procedure

1. Select one projected or auxiliary view.

2. Right-click and choose Add circular detail view.

3. Click the region center in the source view and specify radius by clicking or entering paper millimeters.

4. Change Scale factor (relative to actual size) if needed and place the detail away from the source region.

Scale and Annotations

The initial scale is three times the source scale: 1:10 becomes 3:10, and 1:1 becomes 3:1. An entered factor of 3 means 3:1 relative to actual size. Source callout, detail circle, title and scale label are created automatically.

Rename a Detail View

Select the detail and edit Detail name (symbol) in its properties, or use Rename in its context menu. The source callout and title update together. Names must be nonempty and unique within the layout.

Tips

Dimensions show actual model size, not the enlarged size. Details can be moved and rescaled. Section, perspective and detail views cannot be sources for another detail. The circular region retains its creation projection position, so check it after model edits.

Location: Layout	Group: Layout
Layout Section View	Command ID: ViewSection

Overview

Specify a cutting line in a projected view and place the corresponding section on the sheet. The source model remains intact.

Procedure

1. Select the reference projected view.
2. Right-click and choose Section view.
3. Enter a cutting line with at least two points and press Enter.
4. Move to the section placement position, check arrows and viewing direction, then confirm. Esc cancels.

Section Types

A single cutting segment creates a straight section. A polyline connecting parallel cutting segments creates an offset section. Nonparallel cutting segments create an aligned section, rotated into one section plane.

Generated Graphics

The command adds the cutting line, viewing arrows, identifiers, a section title such as A-A, and section hatching. Placement side changes viewing direction; always check the arrows.

Tips

Specify the cutting line within the source view. After geometry updates, check the cutting position and section contents. Drawing lines and labels are separate from modeling commands that physically cut the solid.

Location: Layout	Group: Layout
Layout Hide and Restore Edges	Command ID: LayoutPatch.HideDrawingEdges

Overview

Hide unwanted edges or projected outlines in one view. The source model and other views are unaffected.

Procedure

1. Select the layout view to edit.
2. Right-click and choose Hide edges.
3. Click lines or select a rectangle. Both visible lines and currently displayed hidden lines can be selected.
4. Press Enter to confirm or Esc to cancel.

Restore Hidden Edges

Use Show all hidden edges from the view context menu to clear its individual edge-hiding settings. Undo also reverses the most recent operation.

Display Differences

This hides only selected lines. Use Show view to hide the entire view, or view display settings to change modes including hidden-line display.

Tips

Independent sheet lines are not view edges. Edge visibility is saved in JDC and applied to printing and 2D export. Inspect the drawing after major model changes.

Location: Layout	Group: Layout
Layout Change Edge Widths	Command ID: LayoutPatch.SetEdgeWidth

Overview

Set paper-space line widths for an entire layout view or selected edges without altering model attributes.

Whole-view Width

1. Select the view and enter Edge width (mm) in its properties.

2. Zero uses the original model's line-width attributes.

Selected-edge Width

1. Enter the desired Selected edge width (mm).

2. Run Change selected edge width from that entry or the context menu.

3. Click edges or select a rectangle and press Enter. Esc cancels.

Setting Priority

Individual-edge settings override the whole-view setting. Assigning zero to an edge restores the whole-view setting for it.

Relationship to Scale

Values are paper millimeters. Changing the view from 1:10 to 1:20 retains the specified printed width. Zooming the layout screen scales the displayed width together with the paper.

Tips

Settings are saved in JDC and apply to printing and 2D export. Check their balance with other lines in print preview.

Part 3 Dynamic Blocks

Free supports creating and editing 2D dynamic blocks and operating completed 3D blocks. Creating or editing 3D face actions and exporting DWG/DXF require STD or higher. Save the editable master in JDC format.

A parameter defines a value or direction to change; an action defines which objects move and how. Placing a distance parameter alone does not stretch geometry. Associate an action such as Stretch with it.

How to use this part

For your first block, work through 01–07 in order. See 10 and 18 for linked actions, and 24–33 for detailed 3D face actions, formulas, and visibility.

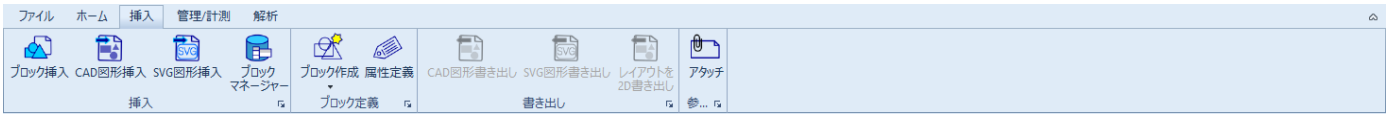


Figure 3-A. Basic operations on the Insert tab. Start from the Block Definition menu.

01 Operate a placed block 274
02 Start creating or editing 275
03 Choose parameters and actions 276
04 Create a distance parameter 277
05 Define the stretch frame 278
06 Select target objects and confirm 279
07 Test actions and save 280
08 Snapping and Tracking 281
09 Configure rotation and flip 282
10 Make a distance parameter follow rotation 283
11 Switch visibility states 284
12 Configure XY and polar parameters 285
13 Configure scaling 286
14 Configure an array 287
15 Create a lookup table 288
16 Change settings later 289
17 Base point and alignment 290
18 Configure chain actions 291
19 Troubleshoot chain actions 292
20 Insert, Explode, and WBLOCK 293
21 Save DWG and DXF with actions 294
22 Settings saved as current geometry 295
23 Mechanical dynamic blocks in Free 296

The geometry, names, and values shown are examples. Use the detailed contents on the next page for 3D face actions, formulas, visibility, and data exchange.

Part 3 Detailed Contents: 3D, Lookup Tables, and Use

Following the basic operations in 01–23, these sections explain how to create and exchange 3D components.

24 Design a 3D dynamic block 297
25 Tutorial: change length by moving an end face 298
26 Register face offset and rotation 299
27 Change hole diameters and copy holes or protrusions 300
28 Move faces on multiple solids with one action 301
29 Use formulas in a lookup table 302
30 Matching size names and maintaining tables 303
31 Coordinate visibility states and grip display 304
32 Choose a save and exchange format 305
33 Dynamic block checklist 306

01 Operate a placed block

Selecting a supported block displays its grips and the Dynamic Block section on the right. Enter distances and angles numerically, and choose visibility states or lookup values from the available options.

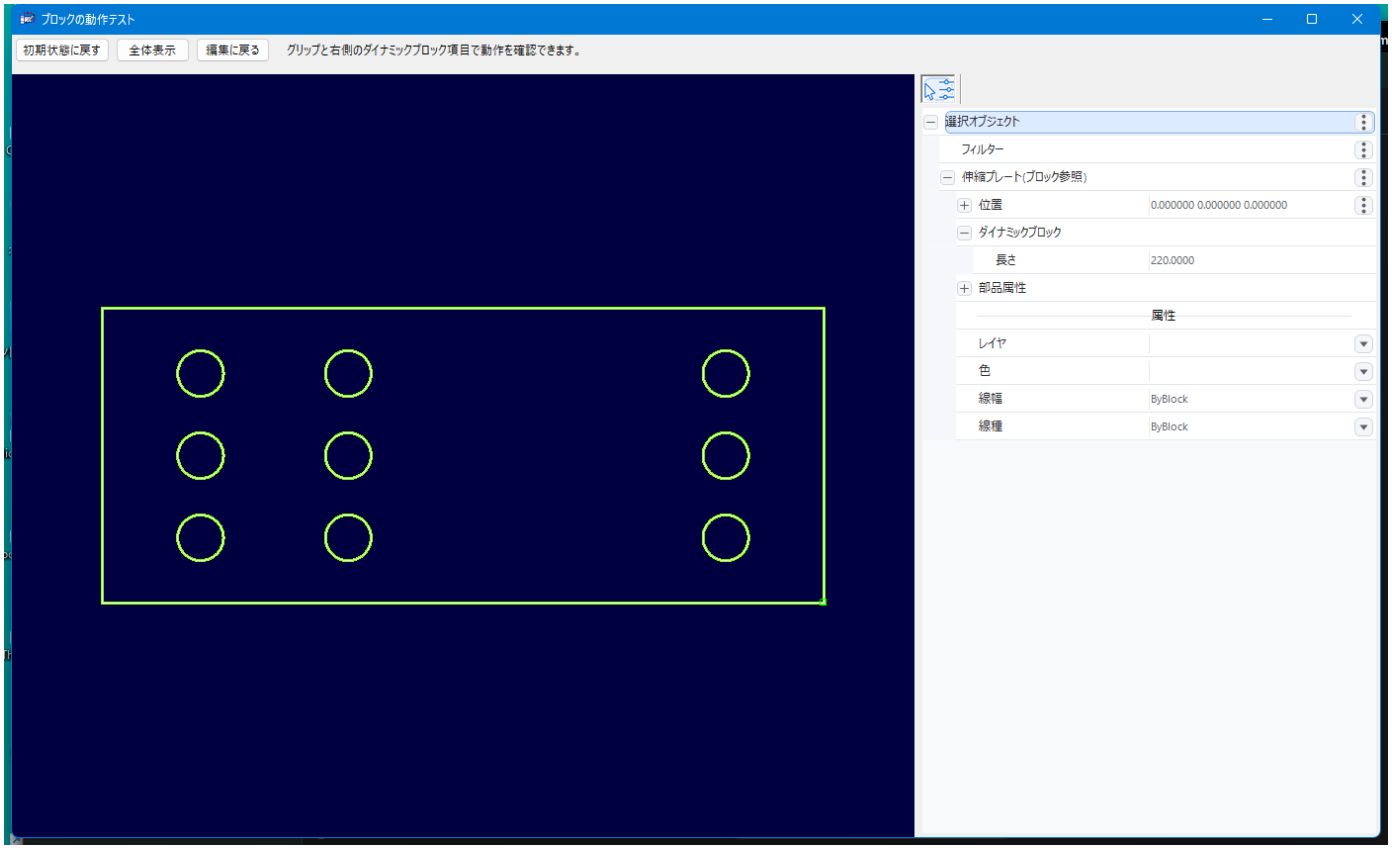


Figure 3-1. Length changed during an action test. Check the value in the Dynamic Block section on the right.

1. Click the block and check the parameter names on the right.

2. Enter and confirm a length or angle, or drag the corresponding grip.

3. Use Undo to reverse a change. This is different from restoring the definition's initial values.

Imported AutoCAD blocks

Modified anonymous blocks may be imported with appearance preservation taking priority. If the context menu offers Reset to initial state and start dynamic editing, that operation resets the placement. Save a separate JDC copy first if you need to retain the current dimensions and geometry.

Definitions containing unsupported settings or constraints may not be fully editable. Even if they look like ordinary blocks, check which actions are supported.

02 Start creating or editing

To create a block, select the source objects and choose Insert > Block Definition > Create Block ▼ > Create Dynamic Block…. You can also start without a selection and draw inside the editor.

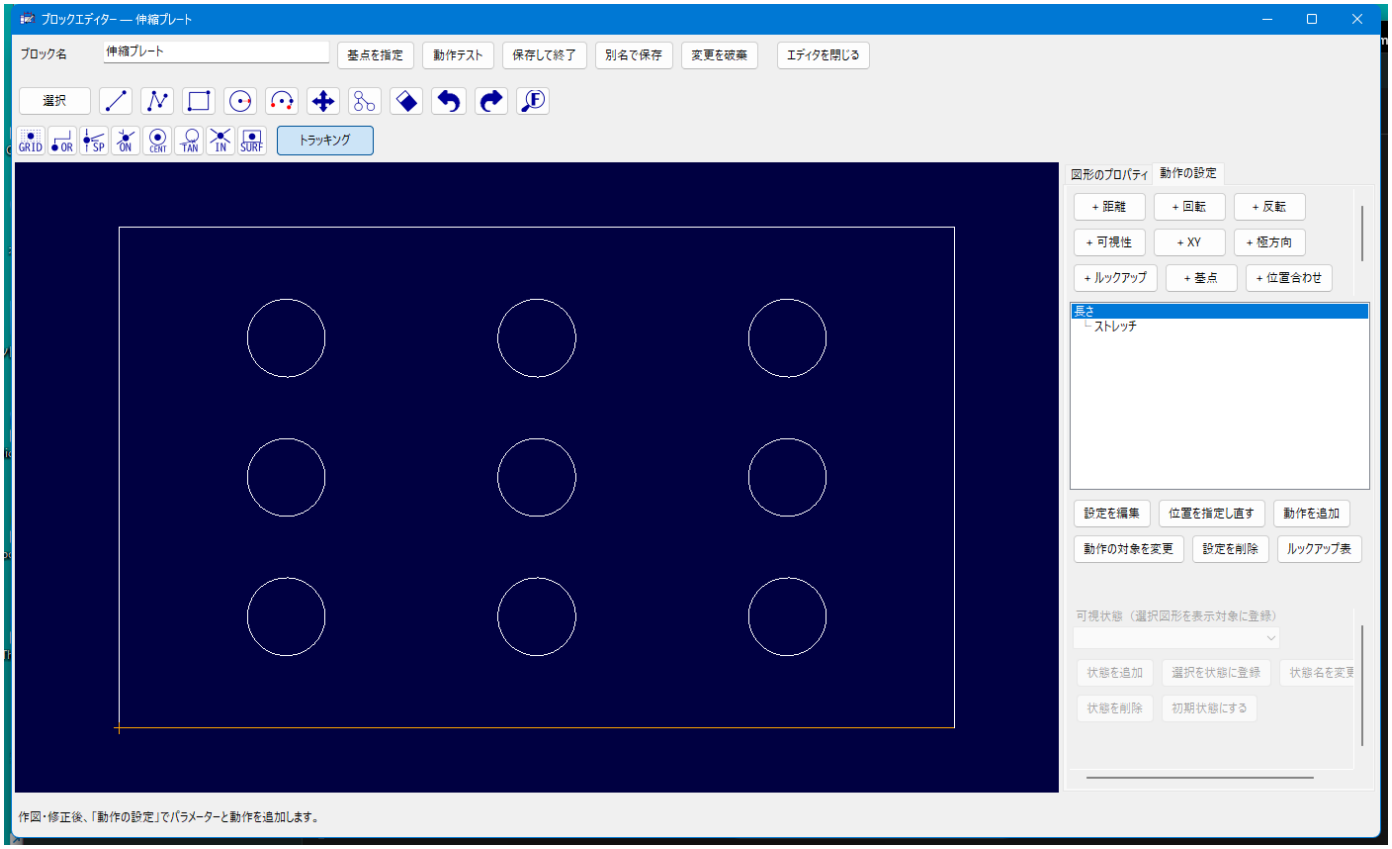


Figure 3-2. Block editor. Drawing and saving controls are at the top; parameters and actions are configured on the right.

To edit an existing block, select it, right-click, and choose Edit Block Definition…. If no placement is selected, choose from the list of named definitions.

Roles of the editor panels

Object Properties controls object attributes. Action Settings controls parameters such as distance and their associated actions. The editor shows objects that visibility states may hide. Use Test Actions to check the actual appearance.

The editor is a separate workspace from the original drawing. Choose Save and Exit to apply the definition to the drawing, then save the drawing file as well.

03 Choose parameters and actions

Add parameters that suit the required behavior. 2D actions primarily use the block's XY plane; 3D face actions use directions and axes in XYZ. See section 24 onward for details.

Parameter	Main use	Associated operations
Distance	Change length or spacing	Move, Stretch, Uniform Scale, Array
Rotate	Change an angle	Rotate
Flip	Mirror left/right or another direction	Flip
Visibility	Switch which geometry is displayed	Register objects for each state
XY	Change horizontal and vertical extents	Move, Stretch, Array
Polar	Change distance and direction	Polar Stretch, Move, Rotate, Scale, Array
Lookup	Select multiple values by size name	Register a table of numeric values
Base Point	Define and move the insertion reference	Configure following of other actions
Alignment	Align with a line or arc	No separate action needs to be added
Position	Change the position of one point	Link to Move or other actions

Common settings

Display name is the name shown in the placed block's properties. Use Minimum, Maximum, Increment, and Allowed values to restrict input. Separate allowed values with semicolons, for example 100;150;200. Leave unrestricted fields blank.

Show in properties and Show grip are separate settings. Turn off grip display if the parameter should be controlled only by numeric input.

04 Create a distance parameter

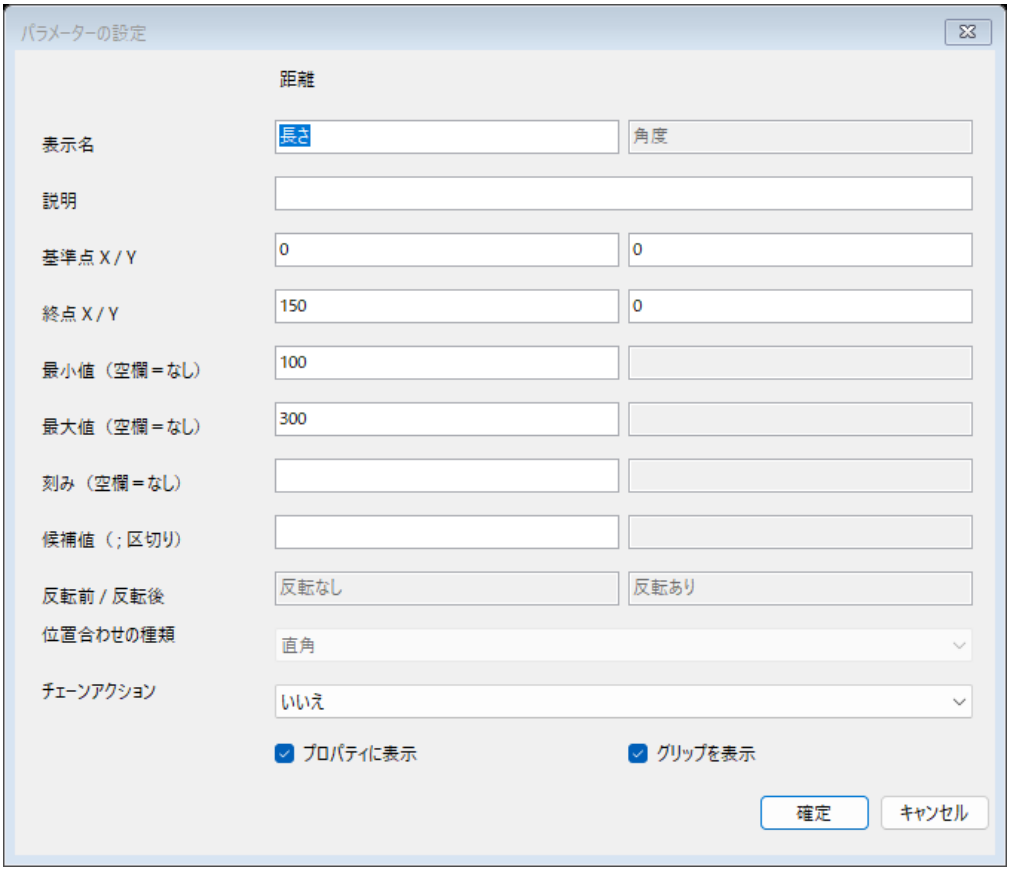


Figure 3-3. Distance parameter settings. For a simple stretch, Chain actions can remain No.

1. Choose Action Settings > + Distance.
2. Click the base point, then the endpoint. Choose points that match the dimension to change, such as circle centers or rectangle endpoints.
3. Set a display name such as Length, enter any limits, and confirm.
4. Check that the new parameter appears in the list on the right.

The example has an initial length of 150, limited to 100–300. Set Chain actions to Yes when this parameter must receive changes from another parameter.

05 Define the stretch frame

1. Select the distance parameter in the list on the right. The orange line on the geometry indicates the selected parameter's location.

2. Choose Add Action > Stretch and confirm the settings.

3. Click two opposite corners of the stretch frame. To extend the right end, include only the moving right end and leave the fixed left end outside the frame.

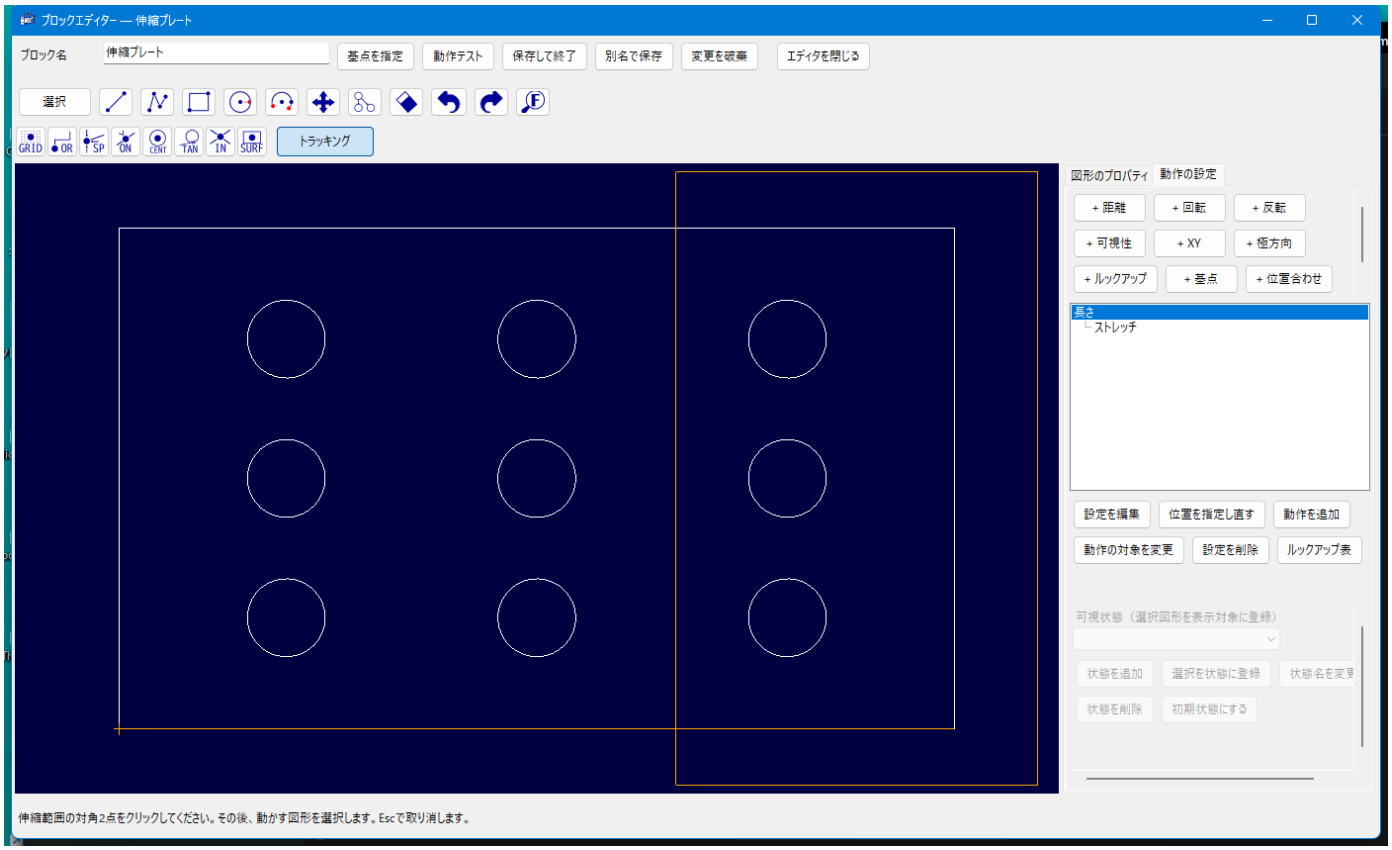


Figure 3-4. An orange stretch frame. After defining the frame, select the objects affected by the action.

Stretch frame versus target objects

The frame defines which endpoints move. The subsequent target selection defines which objects are affected. Defining the frame alone does not complete registration.

Endpoints of lines and straight polyline segments inside the frame stretch. An object wholly inside the frame moves as a whole. Entire curves such as circles can move, but partial stretching across a curve is not supported.

06 Select target objects and confirm

After defining the frame, wait for the status bar to prompt for target selection, then click the objects or drag a selection window around them.

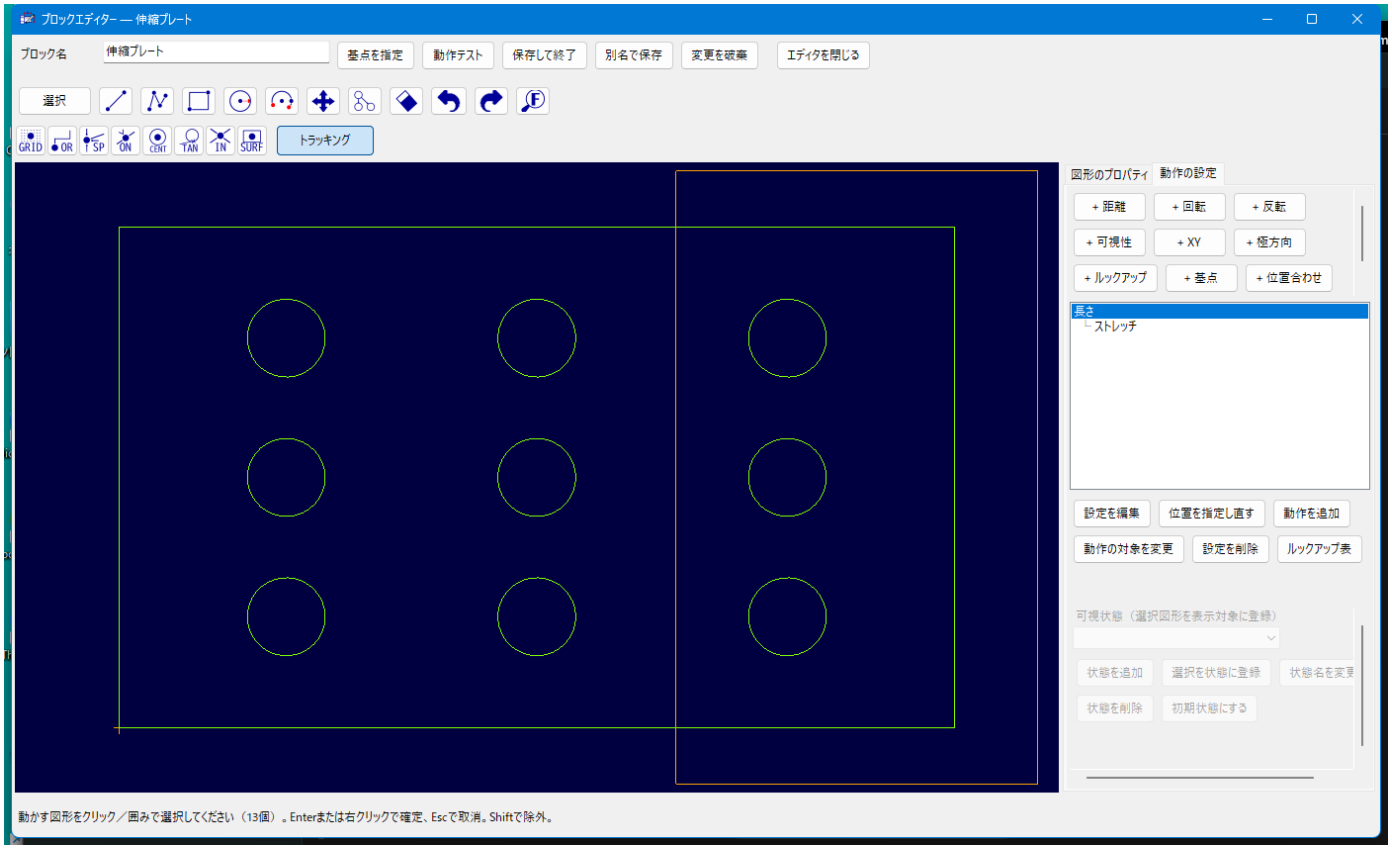


Figure 3-5. Selecting target objects. Even when the whole object is selected, endpoints outside the stretch frame remain fixed.

1. A right-to-left window also selects objects crossing the window boundary. A left-to-right window selects only fully enclosed objects.

2. Click a selected object to remove it. Hold Shift while dragging a window to remove several objects.

3. Check the selection count and confirm with Enter or right-click. Esc cancels.

For a simple stretch, you do not need to select the distance parameter again at this stage. To make another parameter follow, specify it under Target parameters in the action settings.

Check the target count at the bottom and the prompt to confirm with Enter or right-click.

07 Test actions and save

Test Actions lets you try grips and numeric input before applying the definition to the original drawing.

1. Click Test Actions and select the target block.

2. Drag a grip or change a value in the Dynamic Block section on the right.

3. Reset to Initial State reverses changes made during the test. Choose Return to Editing to continue configuring the block.

4. When ready, choose Save and Exit, then save the drawing in the main CAD window.



Figure 3-6. Save and exit controls at the top of the editor. Action testing also starts here.

Button	Effect
Save and Exit	Save and close the definition; update placements using the same definition
Save As	Register another definition without replacing the original placement
Discard Changes	After confirmation, close without applying edits
Close Editor or ×	Prompt to save if modified; Cancel continues editing

Values specific to each placement are retained if still valid in the revised definition. If they cannot be retained, for example because an allowed value was removed, check the message before saving. Undo in the original drawing reverses the definition and placement changes together.

08 Snapping and Tracking

When specifying points, use the object snaps at the top. Enable Endpoint, Center/Midpoint, Intersection, or other snaps as needed, and check the snap marker before clicking.



Figure 3-7. Snap and tracking controls. Enable the required snap types.

Placing parameters

Snaps are available from the first point of distance, XY, polar, and other parameters. You can change snap settings while specifying the second point. Enable Tracking, hover over a reference point to acquire it, then move in the required direction.

The editor inherits the CAD snap settings when opened and restores the main CAD settings when closed. If a point cannot be acquired, check that the appropriate snap type is enabled and that its marker appears.

Moving grips on placed blocks

Distance and rotation grips use the normal polar tracking settings and angular increment. Guide lines appear relative to the rotation center or distance base point. They disappear on confirmation; Esc cancels the drag.

Grip changes to rectangles and dynamic blocks are undone one confirmed drag at a time.

09 Configure rotation and flip

Rotate

Choose + Rotation, specify the center and a point defining the reference direction, then set the display name and angle limits. Select the rotation parameter in the list, add a Rotate action, and select the objects to rotate. Enter angles and increments in degrees.

Flip

Choose + Flip, specify two points on the flip axis, and name the unflipped and flipped states. Add a Flip action, register its target objects, and check both states in Test Actions.

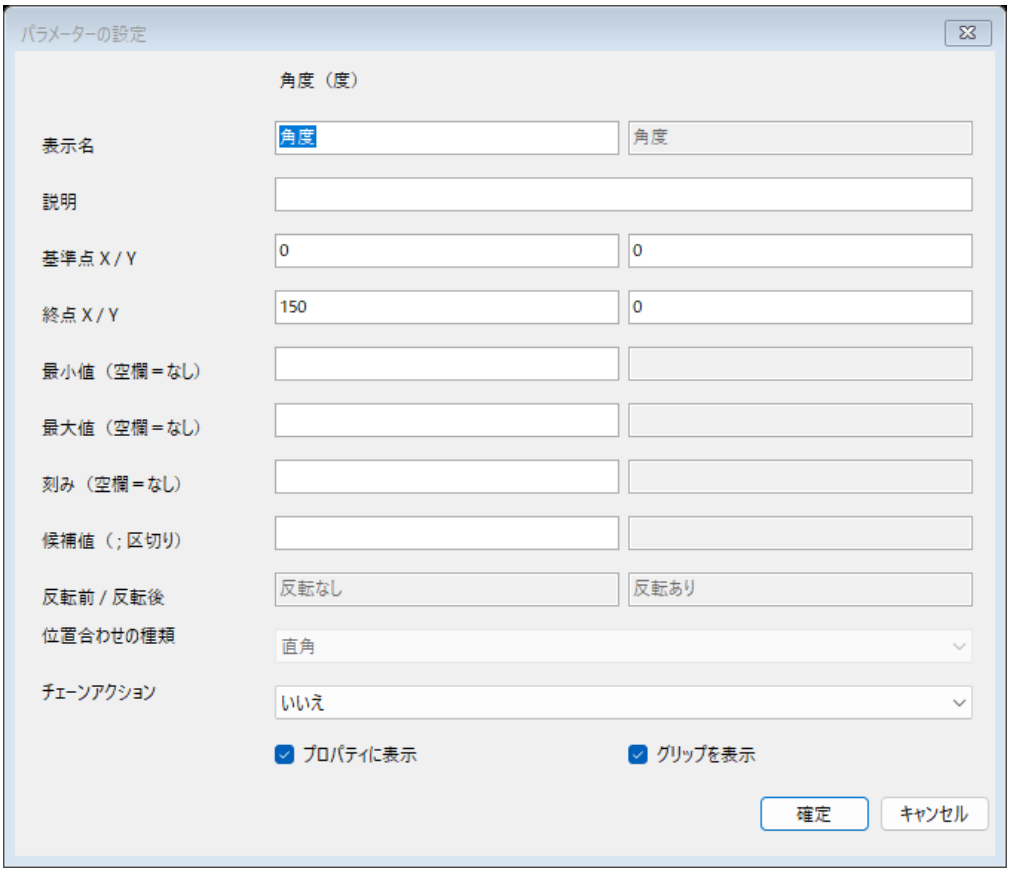


Figure 3-8. Rotation parameter settings. The base point and endpoint define the rotation center and reference direction.

Use Redefine Position to change the parameter location, or Edit Settings to change its name, coordinates, and limits. Choose different target sets depending on whether the whole object or only part of it should rotate.

10 Make a distance parameter follow rotation

If only geometry is targeted by rotation, the distance parameter axis stays in its original direction. To change length along the rotated direction, include the distance parameter as a target of the rotation action.

1. Assign a Stretch action to the distance parameter.
2. Select the rotation parameter's action in the list and open Edit Settings.
3. Under Target parameters, select the distance parameter. Also register the geometry to rotate in the rotation action.
4. Change the angle in Test Actions and check that the distance grip points in the same direction as the geometry.
5. Change the distance in this state and check that stretching follows the rotated axis.

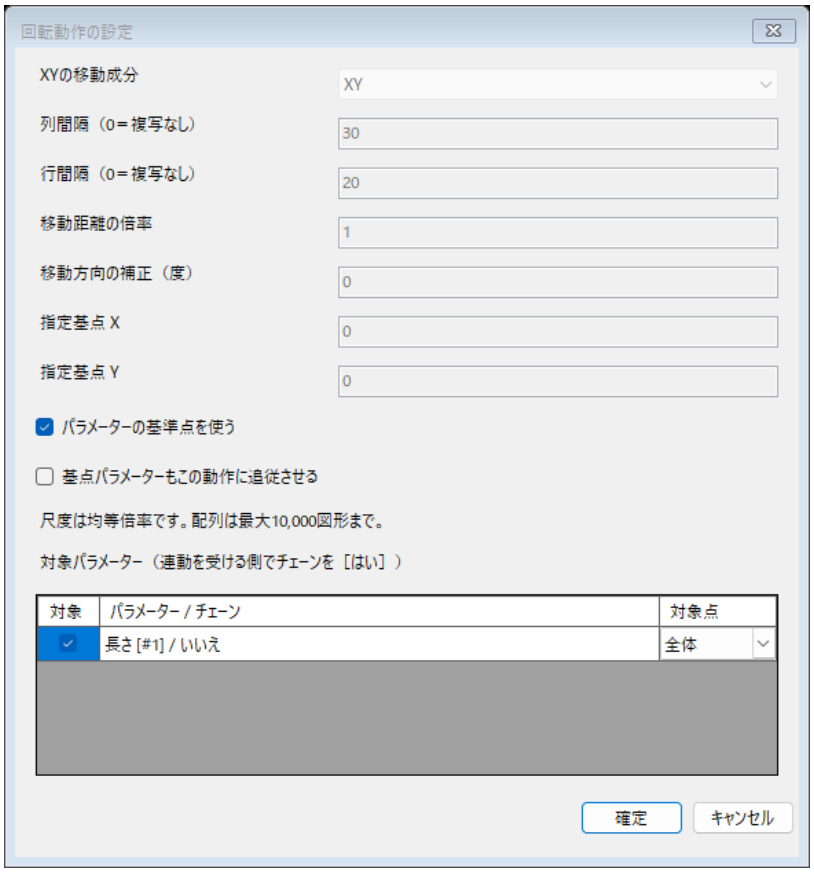


Figure 3-9. Including a distance parameter as a target of a rotation action.

When rotation turns both ends of a distance parameter by the same angle, its length does not change. Following a position or direction is different from a chain action that changes a value and runs a child action.

11 Switch visibility states

Visibility switches between shapes prepared within one block, using state names such as Closed, Half Open, and Open.

1. Draw the objects needed for each state, then use + Visibility to place the switching grip.

2. Select the visibility parameter in the list.

3. Select the objects to display in a state, choose Add State, and enter its name.

4. Change the selection and register the other states. Include shared objects in every relevant state.

5. Switch states in Test Actions to verify the result.

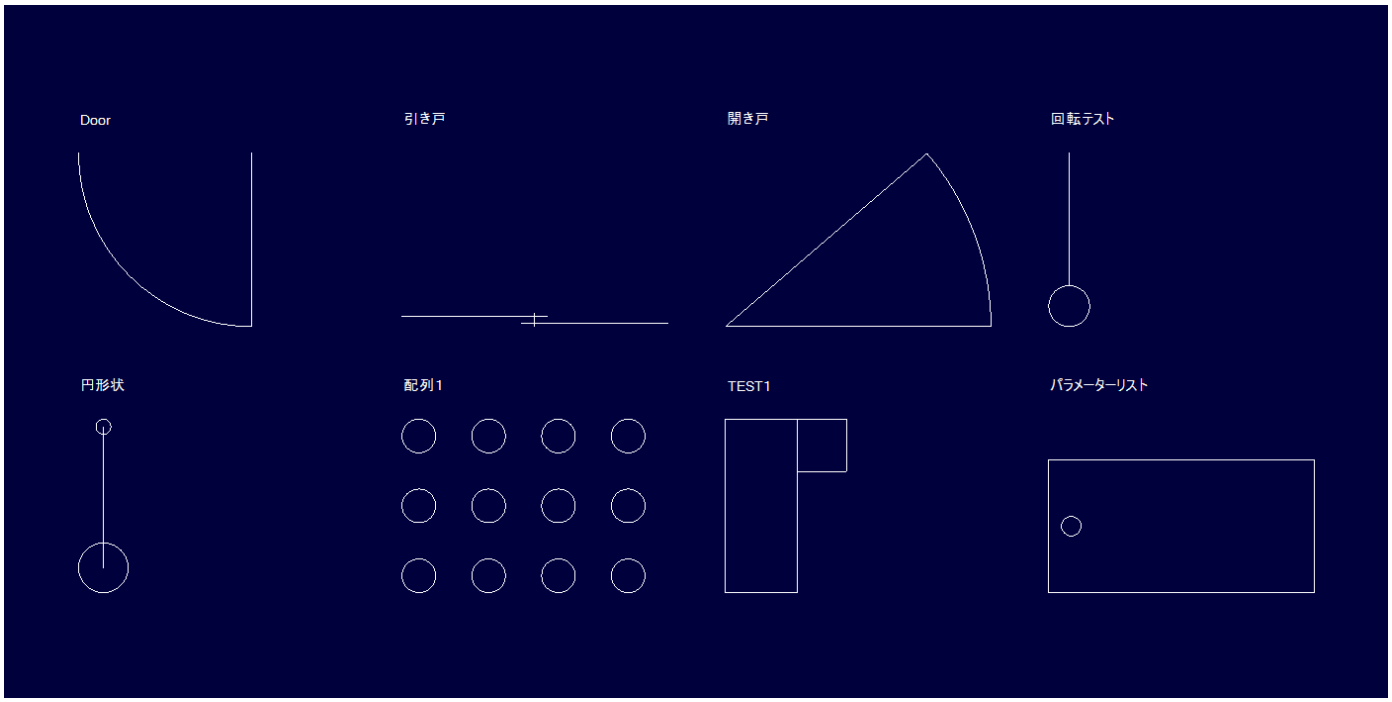


Figure 3-10. An imported dynamic block. A definition with supported visibility settings displays the objects included in the selected state.

To update an existing state, reselect its objects and choose Register Selection to State. You can also rename, delete, or Set as Initial State. There is one visibility parameter; no Move or Rotate action needs to be added.

Registering a state with no selection hides all objects managed by that visibility parameter. Since the editor shows all objects, check the actual state in Test Actions.

12 Configure XY and polar parameters

XY

Choose + XY, specify the base point and endpoint, and set display names and limits for X and Y. The left and right fields in the settings dialog correspond to X and Y.

In Move or Stretch action settings, choose X, Y, or XY under XY movement component. Register width-changing objects with the X action and height-changing objects with the Y action to control both dimensions through one parameter.

Polar

Choose + Polar, specify the base point and endpoint, and set the distance and angle. Angles are in degrees. Polar Stretch moves endpoints inside the rectangle according to the change in distance and angle.

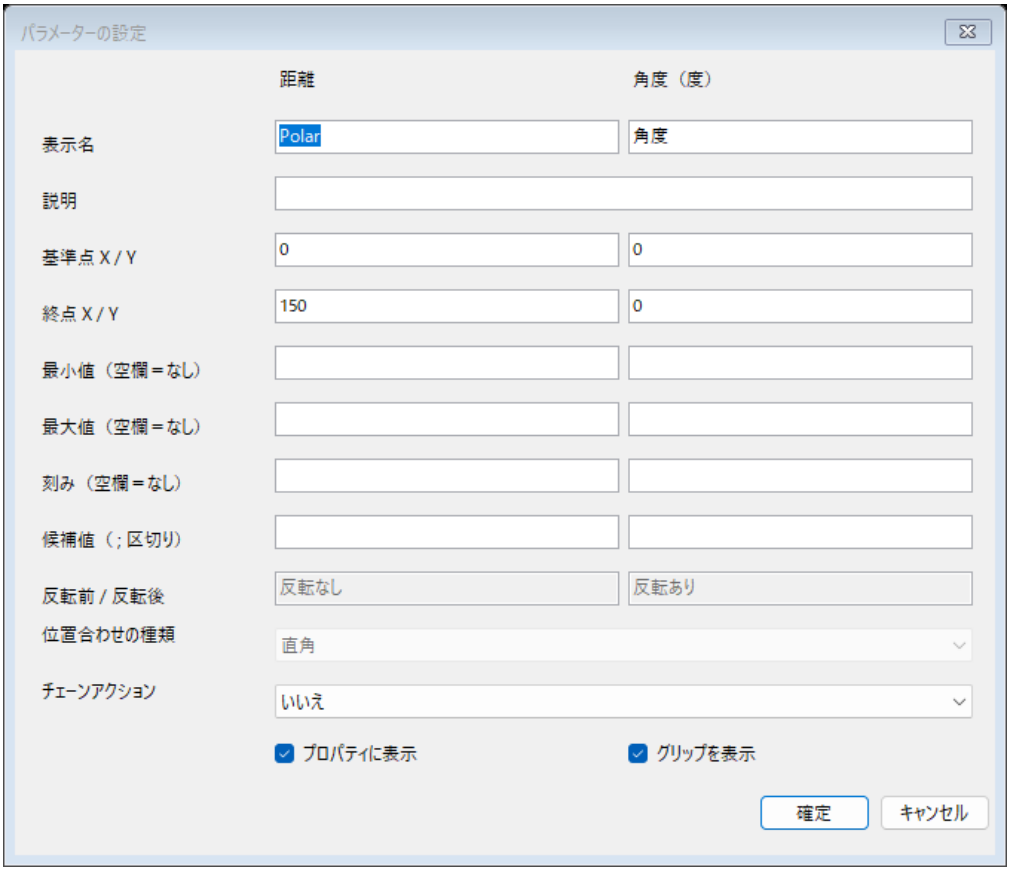


Figure 3-11. Polar distance and angle settings.

Objects wholly inside the rectangle move as a whole. AutoCAD's rotation-only selection set for polar stretch is not supported. Some settings, such as negative XY distances, work in JDC but cannot be saved with actions in DWG/DXF.

13 Configure scaling

A Scale action assigned to a distance parameter scales target objects uniformly by the ratio of the new distance to the initial distance.

1. Use + Distance to specify two reference points. For an initial distance of 100, use points 100 units apart.

2. Select the parameter in the list and choose Add Action > Scale.

3. Check the base point. Use the parameter's base point or specify base-point X and Y coordinates.

4. Select the objects to scale and confirm with Enter or right-click.

5. In Test Actions, a distance of 200 gives a scale of 2; a distance of 50 gives a scale of 0.5.

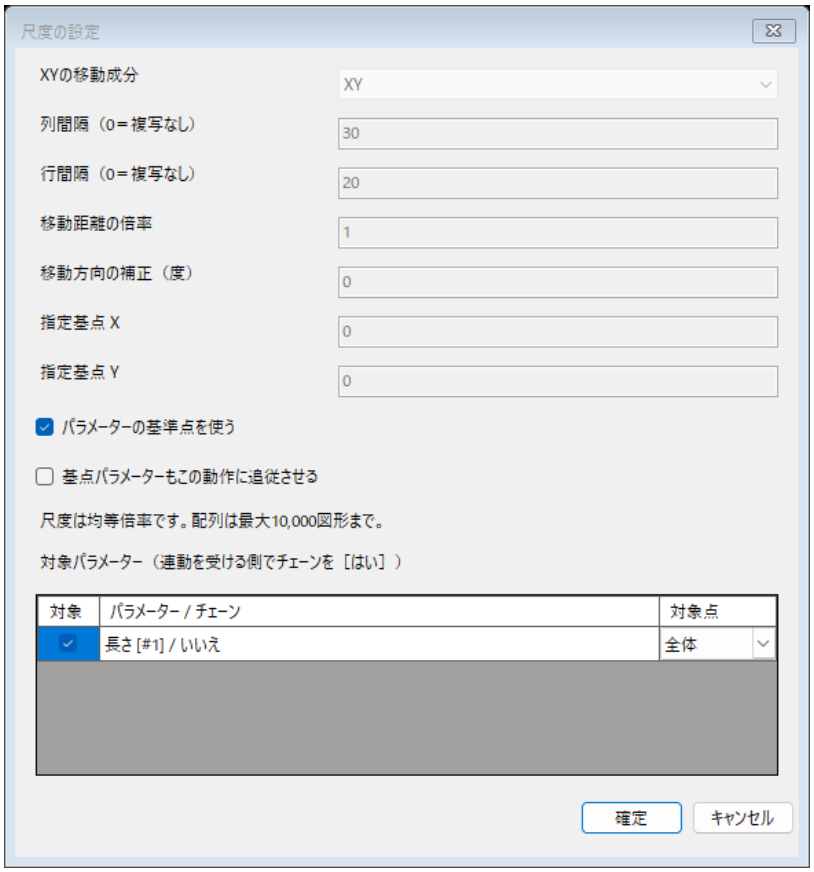


Figure 3-12. Setting the scale base point and target parameters.

Checking the base point and targets

The base point remains fixed during scaling. To enlarge a component while keeping its mounting position fixed, use that mounting position as the base point.

To scale other parameters by the same ratio, register them as target parameters in the action settings. Enable following for the base-point parameter only when needed.

Uniform scaling is supported. Nonuniform scale actions with separate X and Y factors are not supported.

To save actions in DWG/DXF, choose AutoCAD 2018 or later format and check the message shown after saving.

14 Configure an array

An Array action increases or decreases repeated objects at fixed intervals according to a distance or XY range. Use it for repeated features such as holes or bolts.

1. Create a Distance, XY, or Polar parameter.

2. Choose Add Action > Array and specify column and row spacing.

3. Select the source objects to repeat and confirm with Enter or right-click.

4. Change the range in Test Actions and check the column and row counts and positions.

Setting	Copy direction and count
Distance or Polar	Specify column spacing; set row spacing to 0
XY	Specify column and row spacing; 0 disables copying in that direction
Count calculation	Including the source: integer part of range ÷ spacing + 1 in each direction
Example	X = 30, Y = 20, and spacing = 10 in both directions gives 4 columns × 3 rows

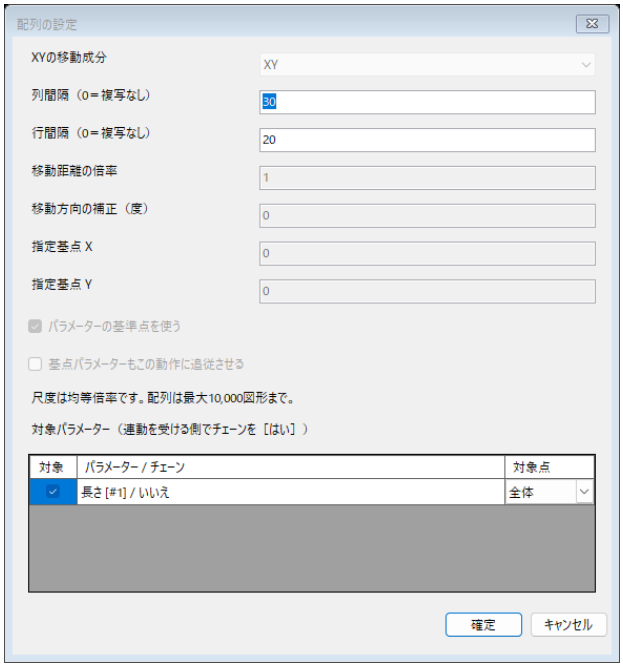


Figure 3-13. XY array action settings for column and row spacing.

A change producing more than 10,000 objects in total is not performed. DWG/DXF may save current geometry for cases such as negative XY distances or an XY array with zero spacing in one direction.

15 Create a lookup table

Select a size name to change several distances or angles together. First register the dimension parameters and their corresponding actions.

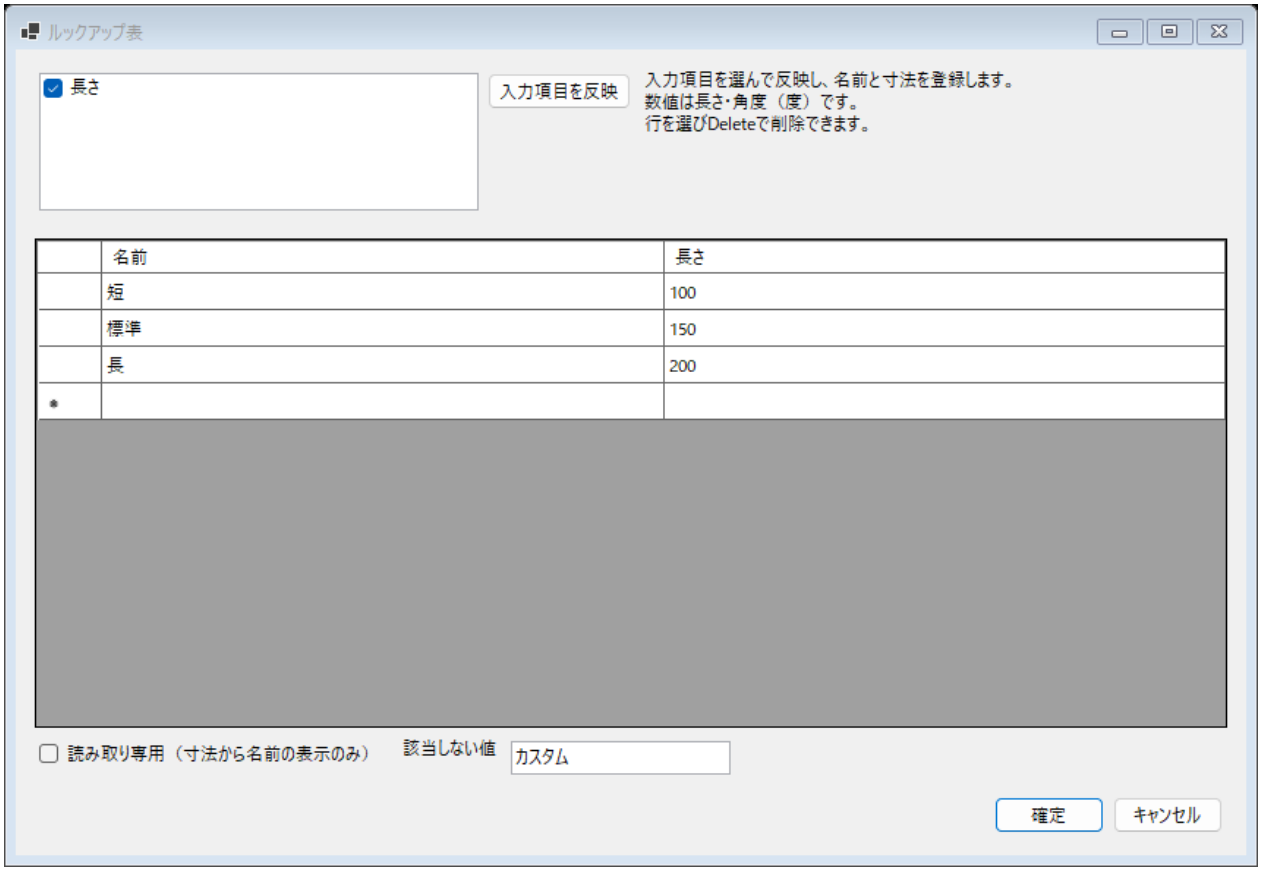


Figure 3-14. Short, Standard, and Long registered with lengths 100, 150, and 200.

1. Use + Lookup to set the grip position and display name.

2. Select the input fields at the top of the table and click Apply Input Fields.

3. Enter a name and corresponding dimensions or angles in each row. Angles are in degrees.

4. Confirm, then select a name in Test Actions.

If dimensions are edited directly, the name of a matching row is displayed. If no row matches, a label such as Custom appears. Read-only mode displays a name based on dimensions, but does not allow selecting a name to change those dimensions.

Values that violate allowed values or limits, duplicate selection names, and duplicate rows with ambiguous matching cannot be registered. The table uses exact numeric matching; ranges, wildcards, and multiple-output tables are not supported.

16 Change settings later

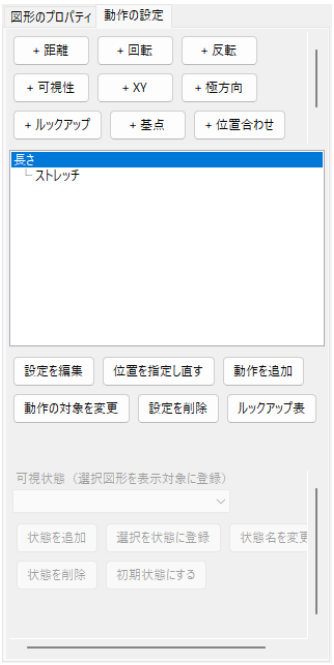


Figure 3-15. List and editing buttons on the right. Distinguish parameters from the actions listed beneath them.

What to change	Operation
Display name, coordinates, limits, chain actions	Select the parameter > Edit Settings
Spacing, multiplier, base point, linked targets	Select the action > Edit Settings
Objects to move	Select the action > Change Action Targets, then confirm the selection
Parameter placement	Redefine Position
Lookup sizes and dimensions	Select the parameter > Lookup Table

To reposition a lookup parameter, select it in the list and drag the orange square that appears. Release the mouse to confirm or press Esc to cancel. Snaps are available.

Changing lookup table rows does not automatically replace valid dimensions on existing placements. To apply revised size dimensions, select the size name again on the placement.

17 Base point and alignment

Insertion base point and action base point

Set Base Point at the top changes the insertion reference. If a base-point parameter exists, its position is changed; otherwise, the ordinary insertion base point is changed.

+ Base Point adds one base-point parameter that can also follow other actions. To move the insertion reference with a distance change, enable Make the base-point parameter follow this action in the action settings.

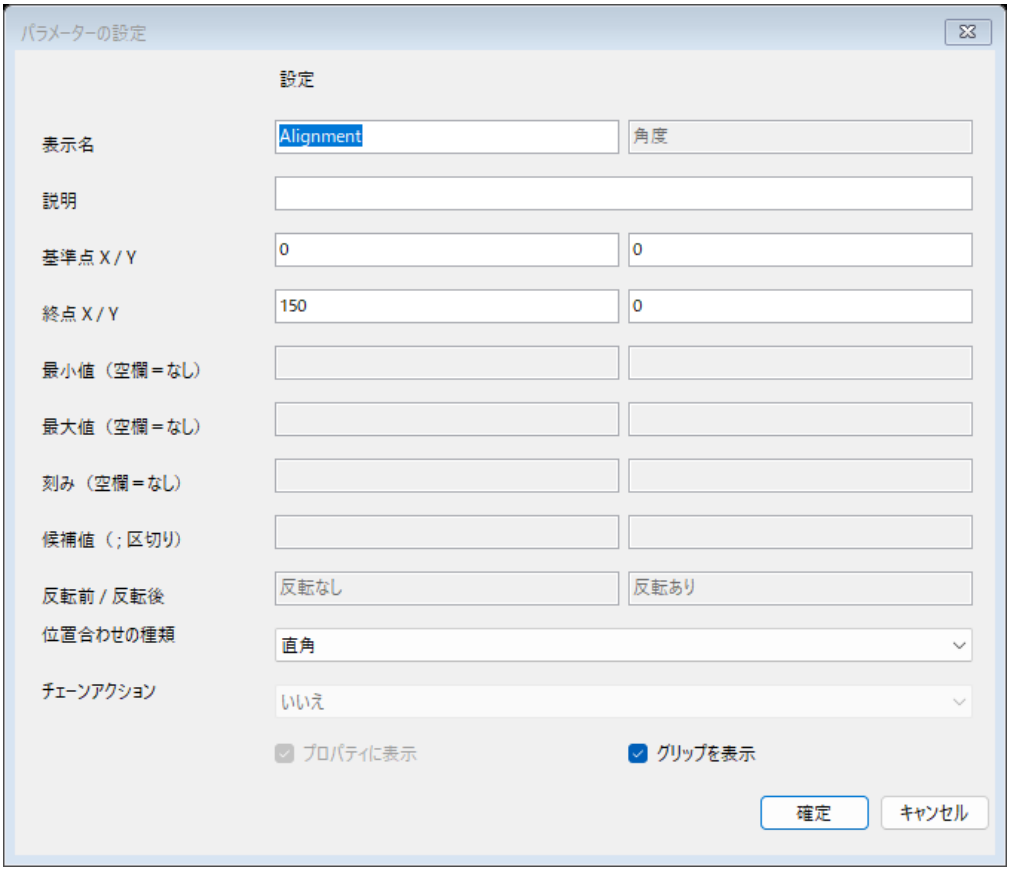


Figure 3-16. Alignment settings: perpendicular or tangential.

Alignment

Use + Alignment to specify a base point and a direction point. During insertion or when dragging the alignment grip, approach a coplanar line, arc, or circle to align the block's position and orientation. Hold Ctrl to reverse the orientation.

Alignment moves the entire block, so no additional action is required. Automatic alignment to arbitrary 3D faces or splines is not supported. Definitions containing alignment are subject to current-geometry-only saving in DWG/DXF.

18 Configure chain actions

Chain actions propagate a change from one parameter to another and execute the latter's associated actions.

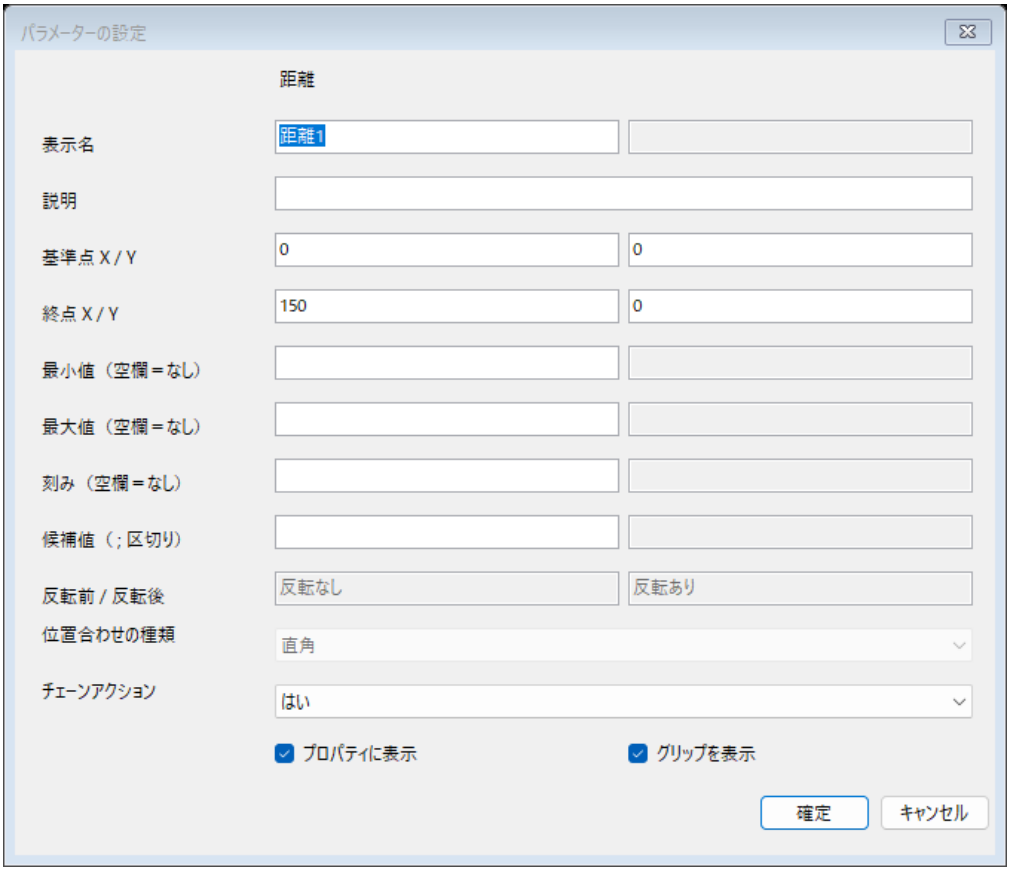


Figure 3-17. Set Chain actions to Yes on the receiving parameter.

1. Select the receiving parameter Distance 1, open Edit Settings, and set Chain actions to Yes.

2. Register an action such as Stretch and the target geometry for Distance 1.

3. Open the action settings for the initiating parameter Distance 2 and select Distance 1 under Target parameters.

4. To change the length of Distance 1, normally set the stretch target point to End point.

5. In Test Actions, change Distance 2 and check that Distance 1 and its target geometry change.

19 Troubleshoot chain actions

Symptom	Settings to check
Distance 1 moves, but its length does not change	Check whether Target point is Entire parameter. To change length, target the start or end point
Distance 1 changes position or value, but its geometry does not move	Check that Distance 1 has Chain actions = Yes and has its own action and target objects
Geometry moves twice	Check that the same object is not targeted by both parent and child actions
The change is canceled	Check the child's minimum, maximum, increment, and other limits
The settings cannot be registered	Check for self-reference or a circular chain

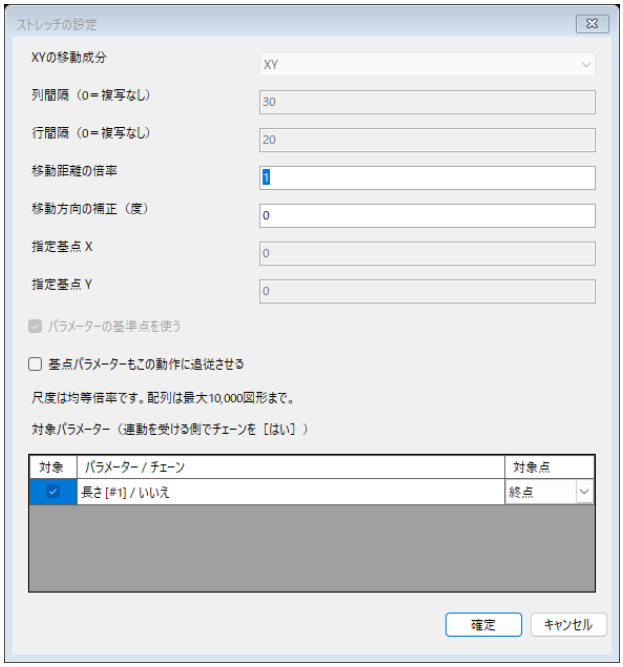


Figure 3-18. Target parameter and target point fields. Use Entire parameter for a Position parameter, which has only one point.

Translating an entire parameter does not change the distance between its endpoints. Check positional following separately from a change in the distance value.

A child with an array updates its count using the resulting distance. Copying the parameter itself through an array is not supported.

20 Insert, Explode, and WBLOCK

Operation	Insertion references and block handling
Insert a registered block	Align the definition's base point with the insertion point
Insert a block from a file	Use the drawing's saved insertion base point, or the origin if none is set
Insert CAD geometry	Place relative to the drawing origin and retain the original block references
Write Block (WBLOCK)	Preserve coordinates and save the specified point as the drawing's insertion base point

Explode on insertion

Choose Explode on insertion below Rotation angle in the block insertion settings. The initial setting is No; later selections are saved when the application exits normally.

When inserting a drawing file containing block references with this option set to Yes, only the outer wrapper level is removed. Internal dynamic block references remain, allowing their grips to be operated directly.

When inserting a registered dynamic block itself with Yes, its current geometry is exploded into lines, circles, and other objects. Nested block references remain.

Explode after placement

Select multiple objects and run Explode to process those that support it. Dynamic blocks are also exploded one level using their current geometry and visibility state, losing their dynamic grouping.

Exchanging drawings with an insertion base point

JDC retains the drawing's insertion base point, which maps to INSBASE in DWG/DXF. This point is separate from the base points of individual block definitions. If a definition with the same name exists at the destination, the imported definition receives a numbered name to protect the existing one.

21 Save DWG and DXF with actions

JDesignCAD 1.1.4 can save supported dynamic blocks to AutoCAD 2018 or later DWG/DXF formats. Both formats support the same scope. Export requires STD or higher.

1. Save the editable master in JDC format.
2. Choose DWG or DXF in Save As or another export command.
3. Select AutoCAD 2018 or later in the output settings.
4. Check the message after saving. Blocks saved as current geometry are listed with their names and reasons.
5. In the receiving CAD application, verify distance, angle, and state changes and new insertion, as well as appearance.

Main features saved with actions	**Details**
Distance and rotation	Supported Move, Stretch, and Rotate actions and grips
Scale and array	Uniform scaling and supported Distance, XY, and Polar arrays
Visibility and flip	Visibility states and unflipped/flipped states
Polar Stretch	Supported stretch frames and target objects
Lookup	Distance, angle, and XY selection through supported numeric tables
Base point and linked parameters	Supported base-point following; moving, rotating, scaling, and endpoint stretching of other parameters; chain actions

AutoCAD does not need to be installed for export. Actions lost when an earlier version exported static DWG/DXF geometry cannot be recovered from those files. Open the original JDC and export it again.

22 Settings saved as current geometry

Even after a successful save, blocks with unsupported settings may not allow parameter editing in AutoCAD. This warning does not mean saving failed; it means the block was saved in its current geometric form.

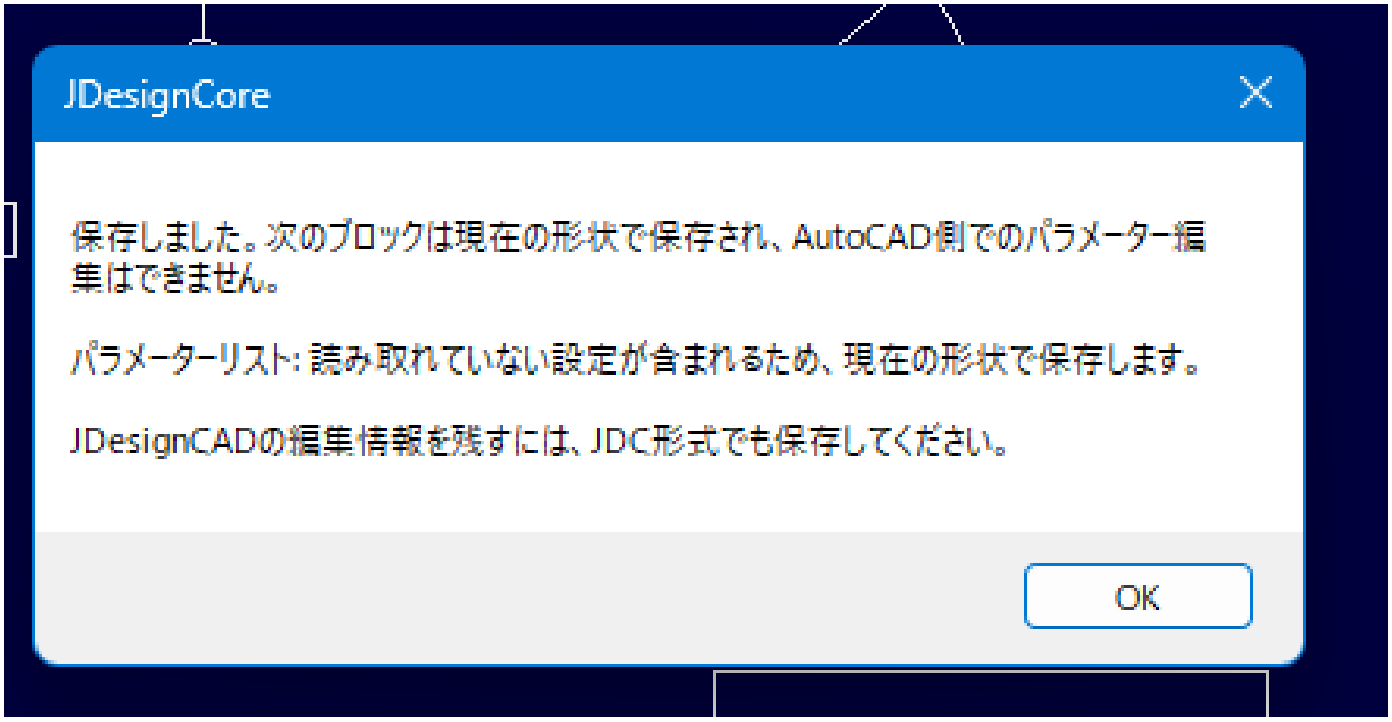


Figure 3-19. Save message for a block containing unreadable settings. Check the target name and reason, and retain the editable JDC source.

The following settings are generally outside the scope of saving with actions.

- Block property tables, geometric and dimensional constraints, and unreadable settings.
- Alignment, independent Move actions, and rotation-only selection sets for Polar Stretch.
- Hiding parameters themselves through visibility; nonuniform scaling; negative XY distances; XY arrays with zero row or column spacing.
- Combinations whose evaluated results do not match, such as independent rotation and an array driven by another parameter acting on the same object.

Full compatibility with every AutoCAD definition and action combination is not guaranteed. For components that must remain editable in the receiving application, test the required operations before delivery. Retain the JDC master for further editing in JDesignCAD.

23 Mechanical dynamic blocks in Free

Using the eyebolt in test22.jdc, change the length and orientation of the same block. Free also allows supported parameters to be operated and saved to JDC without redrawing the geometry.

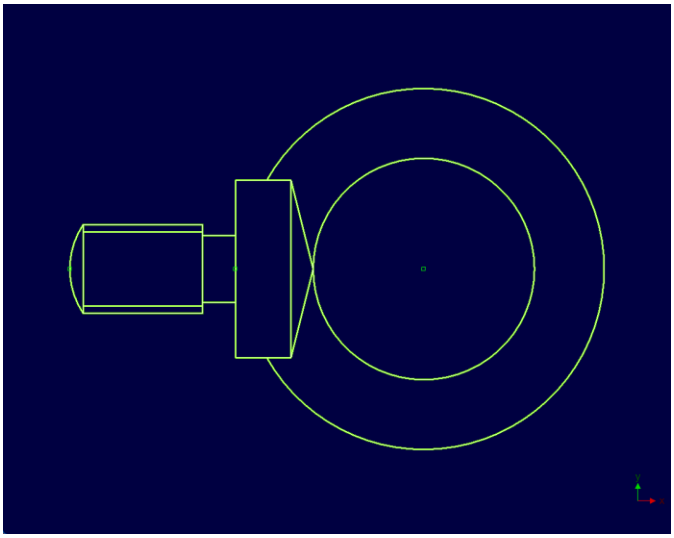


Figure 3-20. Initial state of test22.jdc: distance 12 and rotation 0°.

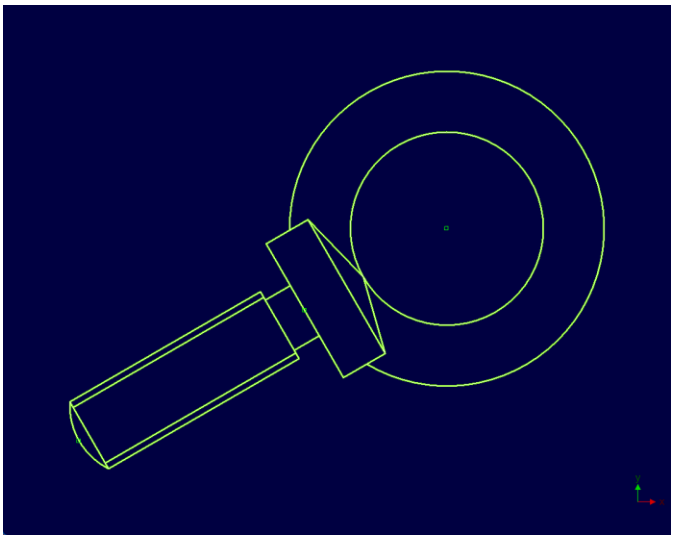


Figure 3-21. Distance changed to 24 and rotation to 30°. The entire object, including the ring and shaft, rotates.

1. Open test22.jdc and select the component's block reference.

2. Under Dynamic Block on the right, change Distance from 12 to 24.

3. Set Rotation (deg) to 30 and check the geometry and grip orientation.

These changes illustrate operation only; they do not guarantee that the resized component meets a standard or has adequate strength. Keep the original and save to a differently named JDC file.

24 Design a 3D dynamic block

For 3D components, distinguish actions that move the entire solid from those that modify its faces. Move an end face to change only a bar's length; move the whole solid to reposition the component.

Objective	Parameter	Action or setting
Change a shaft or plate's length	Distance	3D Face Move
Change wall thickness or face projection	Distance	3D Face Offset
Tilt a face	Rotate	3D Face Rotate
Create another hole or protrusion	Distance	3D Face Copy
Change a round hole's diameter	Distance	3D Hole Diameter
Select standard dimensions	Lookup	Register dimension columns and any required formulas

Available editions

Creating and editing 3D face actions requires STD or Pro. Free can insert completed blocks, operate supported dimensions, lookup selections, and grips, copy the blocks, and save them to JDC. Free also supports creating and editing conventional 2D dynamic blocks.

Decide before creating

1. Define the initial geometry and dimensions.

2. Distinguish fixed faces, moving faces, and the orientation of the whole component.

3. Set minimum, maximum, and allowed values to prevent invalid dimensions.

4. Start with one action. Test its initial, smaller, and larger values before combining actions.

3D face actions are evaluated from the initial geometry in registration order, followed by whole-component movement, rotation, and other actions. Repeated dimension changes do not accumulate incremental modifications.

25 Tutorial: change length by moving an end face

For this example, create a box inside the block with length 100 mm along X, width 20 mm, and height 10 mm. Keep the X = 0 end fixed and move only the end face at X = 100. This example uses a component defined in millimeters.

1. Select the solid and open Create Dynamic Block…. Set the insertion base point on the fixed end.

2. Check the view and work plane, then use + Distance to specify two reference points along X. Name it Length, set its initial distance to 100, and use allowed values 80;100;150 for this example.

3. Select Length in the list and choose Add Action > 3D Face Move. Set the direction to block-coordinate +X and the multiplier to 1.

4. Select the end face at X = 100, check the selection count, and confirm with Enter or right-click.

5. In Test Actions, choose 80, 100, and 150 in turn. Check that the fixed end stays in place and only the opposite end moves.

6. Choose Save and Exit, then save the original drawing as JDC.

Parameter value	Difference from initial value	End-face movement	Resulting length
80	−20	20 mm in −X	80 mm
100	0	No movement	100 mm
150	＋50	50 mm in +X	150 mm

If movement is not in the expected direction

Movement is based on current value minus initial value, not on the distance value itself each time. Check the direction vector, multiplier sign, and selected face. Entering 150 does not mean moving the face 150 mm.

Dimensions that cannot preserve a closed solid, such as shortening it until a face disappears, are rejected. Check the previous geometry and value, and return to a valid option.

26 Register face offset and rotation

Offset: use each face's outward direction

Register 3D Face Offset with a distance parameter, select the target faces, and confirm. The change from the initial value is applied along each face's outward normal. Even parallel faces move in opposite directions for the same positive increment if their outward normals are opposite.

Action	Direction reference	Use
Move faces	Configured common vector	Move multiple end faces in the same direction
Offset faces	Each face's outward normal	Expand surfaces outward or contract them inward

On a hole's inner wall, outward from the solid points into the hole's void. For a cylindrical hole, 3D Hole Diameter treats the parameter value directly as the diameter.

Rotation: define the axis and initial angle

1. Use + Rotation to specify the center and reference direction, then check the initial angle and allowed values.

2. Choose Add Action > 3D Face Rotate.

3. Set the rotation axis and multiplier, select the target faces, and confirm. The axis uses 3D coordinates within the block.

4. Test small positive and negative angles and check that the intended fixed regions remain in place.

Valid range and action order

Face rotation reconstructs connections to surrounding faces, unlike rigid rotation of the entire component. Large angles, thin walls, and intersections with holes or fillets may fail. Test the required dimension combinations and set limits within the range that produces valid geometry.

Avoid configurations where an earlier action removes a face targeted by a later action. Register actions one at a time and test each before saving.

27 Change hole diameters and copy holes or protrusions

Hole diameter action

Set the distance parameter's initial value to the original hole diameter and associate 3D Hole Diameter. Select the hole feature or a supported cylindrical inner face and test a diameter change. This action uses the distance value itself as the diameter.

Holes at multiple locations created in the same Hole operation change as a group. If one hole must change independently, create it in a separate hole group. Thread standards and thread diameters are not automatically updated.

Face copy action

1. Add a distance parameter and define its initial value.

2. Add 3D Face Copy and set its direction and multiplier.

3. Select a hole, or the side and end faces forming a protrusion, and confirm.

4. In Test Actions, enter a value greater than the initial value. Check that one copy is created while the original feature remains.

Initial distance	Entered distance	Copy position
10 mm	10 mm	Initial state; no additional copy
10 mm	30 mm	20 mm from the original hole or protrusion in the specified direction
10 mm	40 mm	One copy 30 mm from the initial feature

Changing the value repeatedly does not create more copies. This action serves a different purpose from an array of many equally spaced objects.

Selection and geometry requirements

For a protrusion, select all required constituent faces and ensure the copy will connect to the body at its destination. This operation does not copy arbitrary faces as independent surfaces. On solids containing hole-editing information, face copying targets holes. If imported geometry cannot be recognized, consider ordinary hole creation or a different model structure.

28 Move faces on multiple solids with one action

Faces selected from multiple solids in the same block can be registered in one 3D face action. The solids remain separate and retain their colors, layers, and component attributes.

Tutorial: move the boundary between two members

Create two boxes touching at X = 40: member A spans X = 0–40 mm and member B spans X = 40–100 mm. Move only their common boundary to 50 mm while keeping both outer ends fixed.

1. Include both solids in the same block and open the block editor.

2. Create a Boundary Position parameter with initial distance 40 and add 3D Face Move.

3. Set the direction to +X and select A's right end face and B's left end face. Window-select the coincident boundary and check that the target count is 2 faces.

4. Press Enter to register, then set Boundary Position to 50 in Test Actions.

5. Check that A and B are each 50 long, while the overall length remains 100 and both outer ends stay fixed.

Check	Expected result
Joint position	Both end faces share the same X coordinate
Overall length	Remains 100 mm
Structure	Two separate solids retained
Colors and attributes	A and B retain their settings

Operating notes

Face Move, Offset, Rotate, Copy, and Hole Diameter use common target-registration handling, but each solid must satisfy the particular action's requirements. If any target fails, all target geometry and input values remain unchanged.

To configure solids in a nested block, open that inner block. Even if the assembly looks like one object, check whether targets belong to a different definition.

29 Use formulas in a lookup table

A lookup table selects a set of dimensions by size name. Values can be calculated from other columns in the same row. Formulas are saved in JDC.

1. Register parameters such as Length and Lg, together with their actions.

2. Use + Lookup to specify its display name and grip position. Select the table's input fields and click Apply Input Fields.

3. Enter each row's name, Length, and Lg formula. Hover over a cell to check its calculated result.

4. Confirm, then choose a name in Test Actions and check that multiple dimensions change together.

Name	Length	Entry in Lg	Lg result
Size 20	20	Length*0.8	16
Size 30	30	(Length-5)/2	12.5
Size 40	40	=Length*0.8	32

Supported expressions

Use +, −, *, /, parentheses, and positive or negative signs. Multiplication and division symbols × and ÷ and a leading = are also supported. Enclose column names containing spaces or symbols in square brackets, for example [Rotation (deg)]. Use the ordinary ASCII [and] characters and match the column heading exactly.

References may point to columns on the right, but not to other rows. A chain such as Length → Lg → Margin is allowed. Circular references such as Length → Lg → Length, division by zero, unknown column names, and out-of-range results prevent confirmation.

Difference from continuously linked formulas

Formulas calculate table rows. They do not continuously solve every related expression when you directly enter arbitrary dimensions outside the table. Use them to apply consistent dimensions through size selection. If combining lookup selection with free input, check the name-matching behavior in the next section.

30 Matching size names and maintaining tables

Direct numeric changes also display a registered size name if a row matches; otherwise, a label such as Custom appears. Test that the displayed size name agrees with the actual dimensions.

Choose columns used for matching

Use for name matching defines which dimensions identify a row's name. For example, if Length defines the size and Lg is supplementary, exclude Lg from matching. Avoid conditions that cannot distinguish different rows.

Setting or operation	Function
Select a name	Apply the row's dimensions and angles together
Change a number directly	Display the matching row name or Custom
Read-only	Display a name from values; no dimension changes by name selection
Change input fields	Retain values for existing fields; fill blank cells in added columns
Select row headers and press Delete	Delete the selected rows
Redefine Position	Change the lookup grip position

Existing placements after editing the table

Editing a lookup table does not silently replace valid dimensions or angles on existing placements with new size values. Placements that no longer match the table become Custom. To apply a new size, reselect its name in the placement's properties.

If values cannot be retained, for example after removing allowed values, review the message before saving. Overwriting a definition updates placements that use it. Use Save As to create a separate product definition.

Common input errors

Check for duplicate names, ambiguous dimension combinations, empty cells, and results violating minimum, maximum, or allowed values. Formula column names must match the headings exactly. Arbitrary AutoCAD lookup tables, including ranges and wildcards, are not necessarily editable in the same way.

31 Coordinate visibility states and grip display

Switch between plan and side views, or Closed, Half Open, and Open states within one block. Visibility states control object display; Parameter Display controls grip visibility in each state.

Register geometry for each state

1. Create the required objects within the block and add + Visibility. Only one visibility parameter is used.

2. Select the objects to display, choose Add State, and name it, for example Plan.

3. Select other objects and add a state such as Side. Use Register Selection to State to revise an existing state.

4. Use Set as Initial State to choose the state used on insertion. Registering with no objects selected hides all managed objects in that state.

Hide grips unnecessary in a state

1. In Action Settings, select the visibility parameter and state.

2. Open Parameter Display and check only the items to show.

3. Click OK to register, then configure other states as needed.

4. Switch states in Test Actions and on a normal placement, checking both geometry and grips.

Example grips for Plan	Example grips for Side
Width, Depth, plan-view lookup	Height, side-view lookup
Grip for switching visibility states	Grip for switching visibility states

The visibility-switching grip itself cannot be hidden. A grip disabled globally remains hidden in individual states.

Only operating grips are hidden. Property values, lookups, action calculations, and the insertion base point remain active. Older blocks retain their previous grip display settings; use this dialog to hide unnecessary items.

32 Choose a save and exchange format

Format or purpose	What is retained	Recommended use
JDC	3D face actions, formulas, visibility states, and evaluated geometry	Editable master
DWG/DXF with supported 2D actions	Supported dynamic definitions and placements	Verify required actions in the receiving application
Lookup formulas exported to DWG/DXF	Calculated numeric values	Edit the formulas themselves in JDC
DWG/DXF of blocks containing 3D face actions	Current geometry	Action information in the same definition becomes static
Export IFC library	Current visibility state, geometry, and component attributes	See Part 5; action definitions are not transferred to IFC

When to explode a block

Keep reusable component masters as un-exploded JDC blocks. For IFC export, supported blocks containing solids can be selected directly. The solids in the current visibility state are exported as the geometry of one component.

Ordinary Explode extracts the placed geometry as individual objects. If you need to continue using the original dynamic definition, save a separate copy before exploding.

Editing after DWG/DXF import

Anonymous placements modified in another application may be imported with their evaluated appearance preserved. Reset to initial state and start dynamic editing restores the initial state. It does not mean every original definition can be re-edited while retaining the current geometry.

Unsupported constraints, actions, or combinations may be saved as current geometry or excluded from editing. Keep JDC as the master and use exchange formats for delivery, deciding first whether the recipient needs behavior or only appearance.

33 Dynamic block checklist

Symptom	Check first	Next action
Only the parameter moves; geometry is unchanged	Action association and target objects/faces	Add an action or redefine its targets
Stretch setup does not finish	Check whether target selection is pending after frame definition	Select objects and confirm with Enter/right-click
A circle does not deform	Check for partial stretching across a curve	Use whole-object movement or an appropriate dimension/geometry edit
3D geometry extends in the opposite direction	Initial value, direction, and multiplier sign	Test small increases and decreases
3D editing fails at certain dimensions	Thin walls, intersections, or disappearing target faces	Determine the valid range and restrict allowed values
Only one of two solids moves	Check for two selected faces and nested definitions	Check the inner definition and selection count
Unwanted grips remain after switching states	Display settings for each visibility state	Configure Parameter Display
Editing the table does not change placed dimensions	Check whether placement values are being retained	Reselect the size name
Other placements also changed	Distinguish placement-value changes from definition changes	Save a separate definition for a different product
3D actions disappear in DWG	Check whether the definition contains 3D face actions	Provide or retain the editable JDC

Checks before distribution

Test initial, minimum, and maximum values, every lookup row, and every visibility state. Also check translated and rotated placements, copying, Undo/Redo, and saving and reopening JDC files.

Valid geometry and compliance with a product standard are different matters. If dimensions outside the size table are allowed, ensure that the component name alone is not taken as evidence of standards compliance.

Part 4 Analysis

Major STD features include 2D/3D stress analysis, steady-state/transient thermal analysis, beam calculations, trusses, and frames using CAD geometry. Inspect displacement, stress, reactions, strain, and temperature distributions, and use animation and report outputs. These functions are available in STD and Pro.



Figure 4-A. Analysis tab, with icons for the 2D and 3D commands.

01 Register materials 309
02 Prepare analysis geometry 310
03 Basic 2D analysis workflow 311
04 Basic 3D analysis workflow 312
05 Select faces, edges, and vertices 314
06 Partial selection and local meshing in 2D 315
07 Define restraints and loads 316
08 Result quantities and units 317
09 Animate deformation 318
10 Overlay the undeformed outline 319
11 Mesh lines and smooth color display 320
12 Check reactions 321
13 Check strain 322
14 Run steady-state thermal analysis 323
15 Run transient thermal analysis 324
16 Apply convection to all unset boundaries 325
17 Play and save time-step results 326
18 Run beam calculations 327
19 3D beam coordinates and sections 328
20 Trusses and frames 329
21 Export images, printouts, and values 330
22 Save conditions in the drawing 331
23 Example: a component with ribs and holes 332
24 Example: a component with flanges and curved faces 333

Supported calculations

Stress analysis of faces and solids uses small-deformation linear elasticity. Thermal analysis supports steady-state and transient heat conduction. Meshes use eight-node quadrilaterals in 2D and ten-node tetrahedra in 3D.

3D supports multiple materials and perfectly bonded interfaces. Nonlinearity, plasticity, buckling, vibration, separating/sliding contact, coupled thermal stress, automatic self-weight loads, and pass/fail assessment against design standards are not included.

Part 4 Detailed Contents: Setup and Tutorials

In addition to the screen operations in 01–24, these sections cover analysis model selection, multiple materials, and verification examples.

25 Choose the appropriate analysis 334
26 Build the analysis conditions in order 335
27 Configure multiple materials and perfect bonding 336
28 Mesh size and accuracy 338
29 Enter loads correctly 339
30 Tutorial: compare beam results with known values 340
31 3D beam local axes and torsion constant 341
32 Truss/frame connections and results 342
33 Interpret steady-state thermal boundary conditions 343
34 Tutorial: observe cooling in transient thermal analysis 344
35 Distinguish displacement, stress, and reactions 345
36 Save and export conditions and results 346
37 Analysis troubleshooting checklist 347

01 Register materials

Choose Analysis > Basic Settings > Register Material List to add, edit, duplicate, or delete materials. Initial materials include SS400, S45C, SCM440, SUS304, SUS316, SUS430, A5052-H112, A6061-T651, C1100, and C2801.

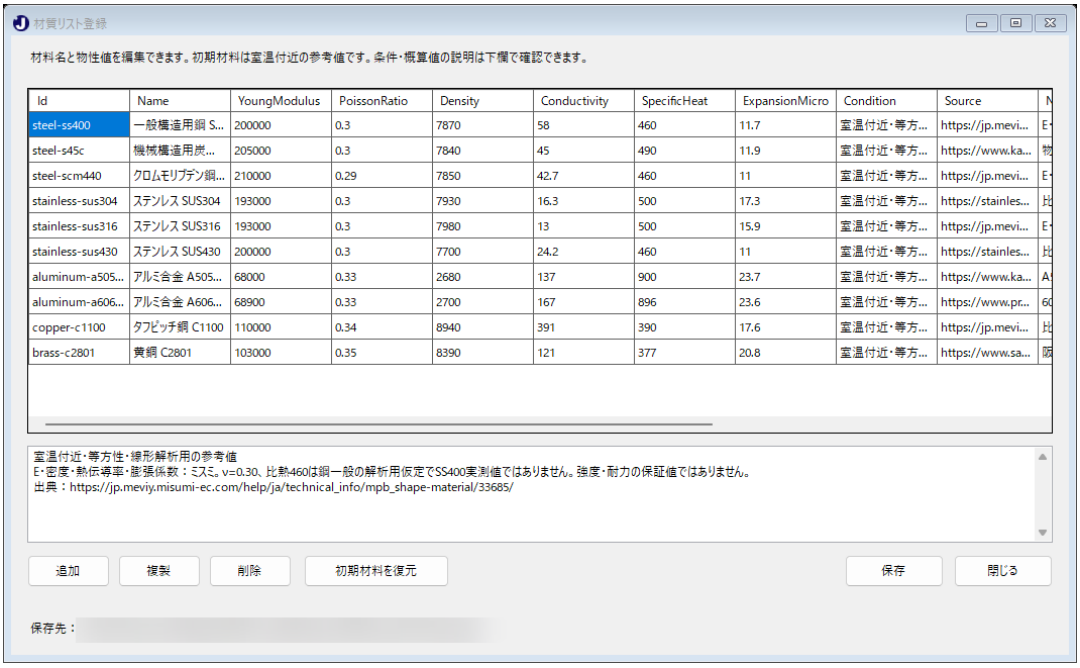


Figure 4-1. Material registration. Select a material and check its properties and the description below. The displayed storage path varies by environment.

Property	Input unit and use
Young's modulus	N/mm²; linear-elastic stiffness
Poisson's ratio	Dimensionless; transverse deformation
Density	kg/m³; used in transient thermal analysis
Thermal conductivity	W/(m K); heat conduction
Specific heat	J/(kg K); heat storage in transient analysis
Coefficient of thermal expansion	10^{-6}/K; can be registered but is not used in current calculations

Initial values are approximate room-temperature references. A material name alone does not guarantee all its properties. Saved analyses retain the material values used at that time. To use updated values from the shared list, reselect the material and recalculate.

02 Prepare analysis geometry

If small details are unnecessary for the analysis, use Optimize Geometry before importing. This modifies the source geometry; make a copy first if the original must be retained.

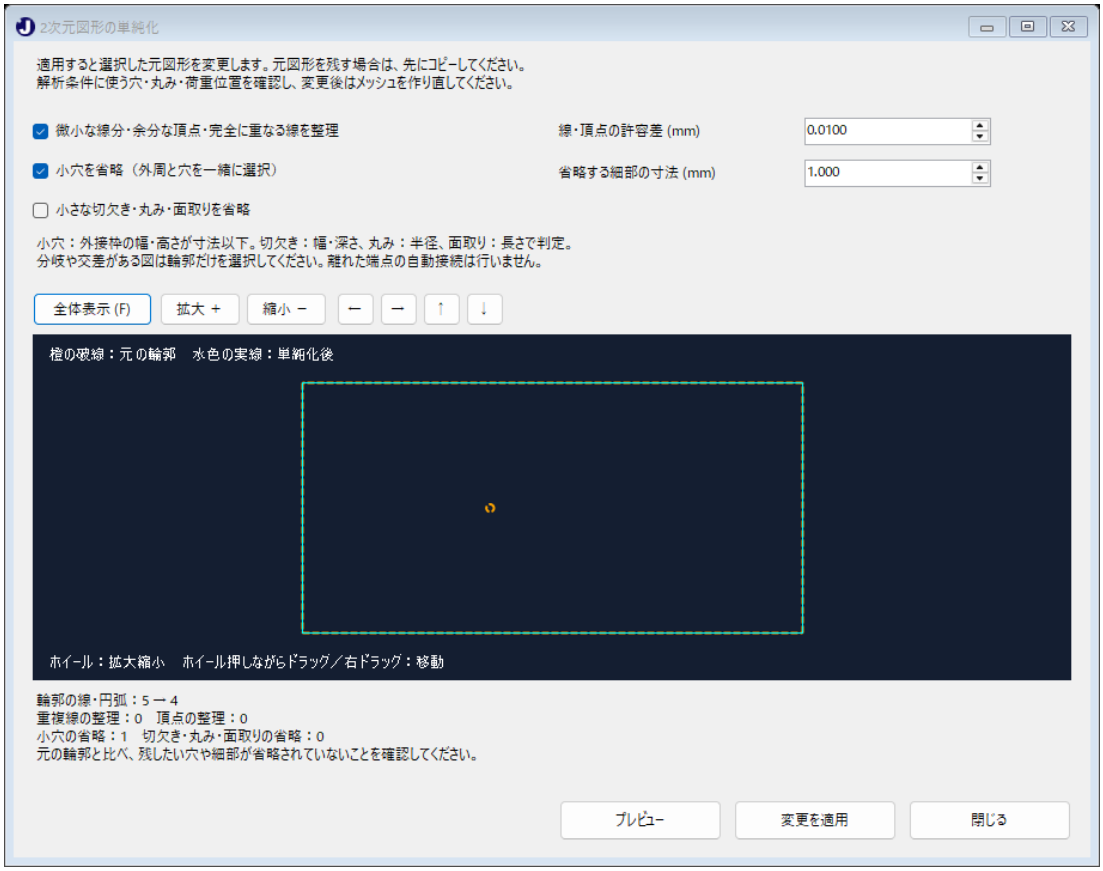


Figure 4-2. Simplifying a 2D outline. Compare the original and proposed outlines in the preview before applying.

1. Select a coplanar outer boundary and holes, or a planar face.

2. Open Optimize Geometry > Simplify 2D Outline > 2D Settings.

3. Select the required operations: clean up lines/vertices, remove duplicate lines, or omit small holes/details.

4. In Preview, compare the orange dashed and light-blue solid lines, then choose Apply Changes.

For 3D, select targets and use the existing Remove Fillet or Delete/Heal Faces commands. This is not an automatic batch operation that detects and removes small holes by size.

Removing load or restraint locations, or holes and rounds whose stress concentrations matter, changes the problem being analyzed. After modifying geometry, import it again and recreate the mesh and conditions.

03 Basic 2D analysis workflow

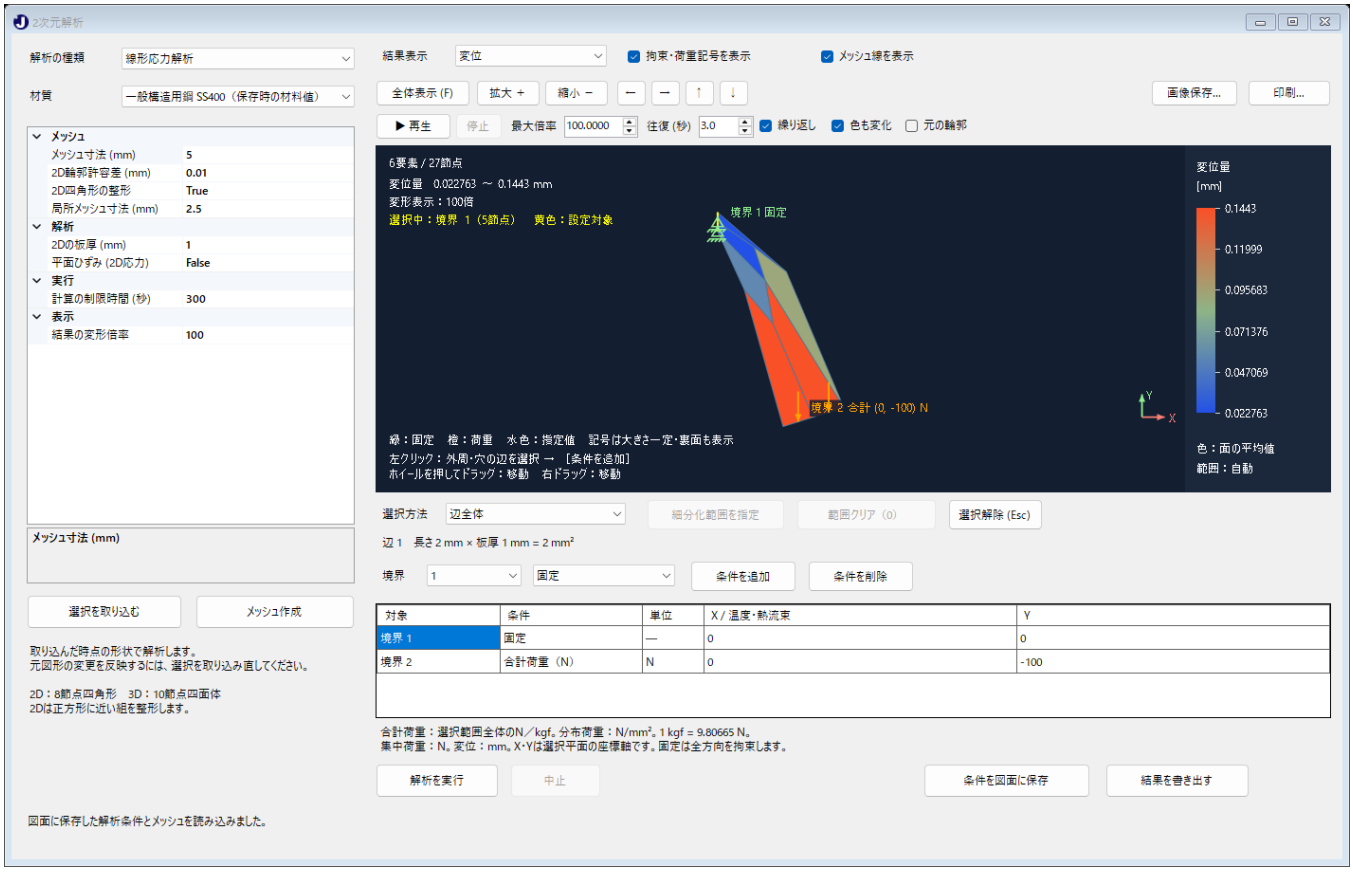


Figure 4-3. 2D analysis window. Configure materials and mesh on the left and define conditions in the table below.

1. In CAD, select one planar face or coplanar outer and hole outlines.
2. Open a Mesh, Stress Analysis, or Thermal Analysis command under Analysis > 2D Analysis, then click Import Selection.
3. Set the analysis type, material, mesh size, and thickness, then click Generate Mesh.
4. Select edges or nodes and click Add Condition for restraints, loads, or other conditions. Enter values in the table and confirm with Enter.
5. Run Analysis and switch between result displays.

For stress analysis, choose plane stress or plane strain. 2D X and Y are coordinates in the imported plane; Z is the thickness direction. Use one outer boundary. Treat disconnected regions and islands inside holes separately.

Changes in CAD do not automatically update the imported geometry. Run Import Selection again after making changes.

04 Basic 3D analysis workflow

For 3D, select 1–100 closed solids. Open 3D Analysis > Import Selection, set materials, bonding, and mesh size, then generate the mesh. See section 27 of this part for multiple materials.

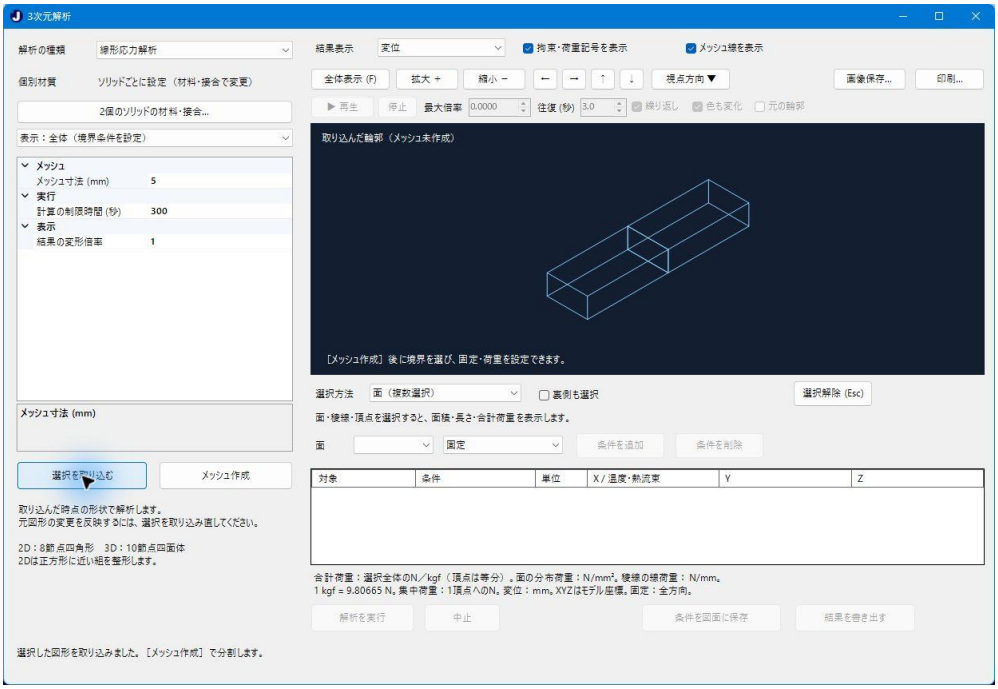


Figure 207 3D Analysis import screen, including materials, contacts and display scope.

The following result figures show single-body examples. Use the current screen above and section 27 for materials, contacts and assembly setup.

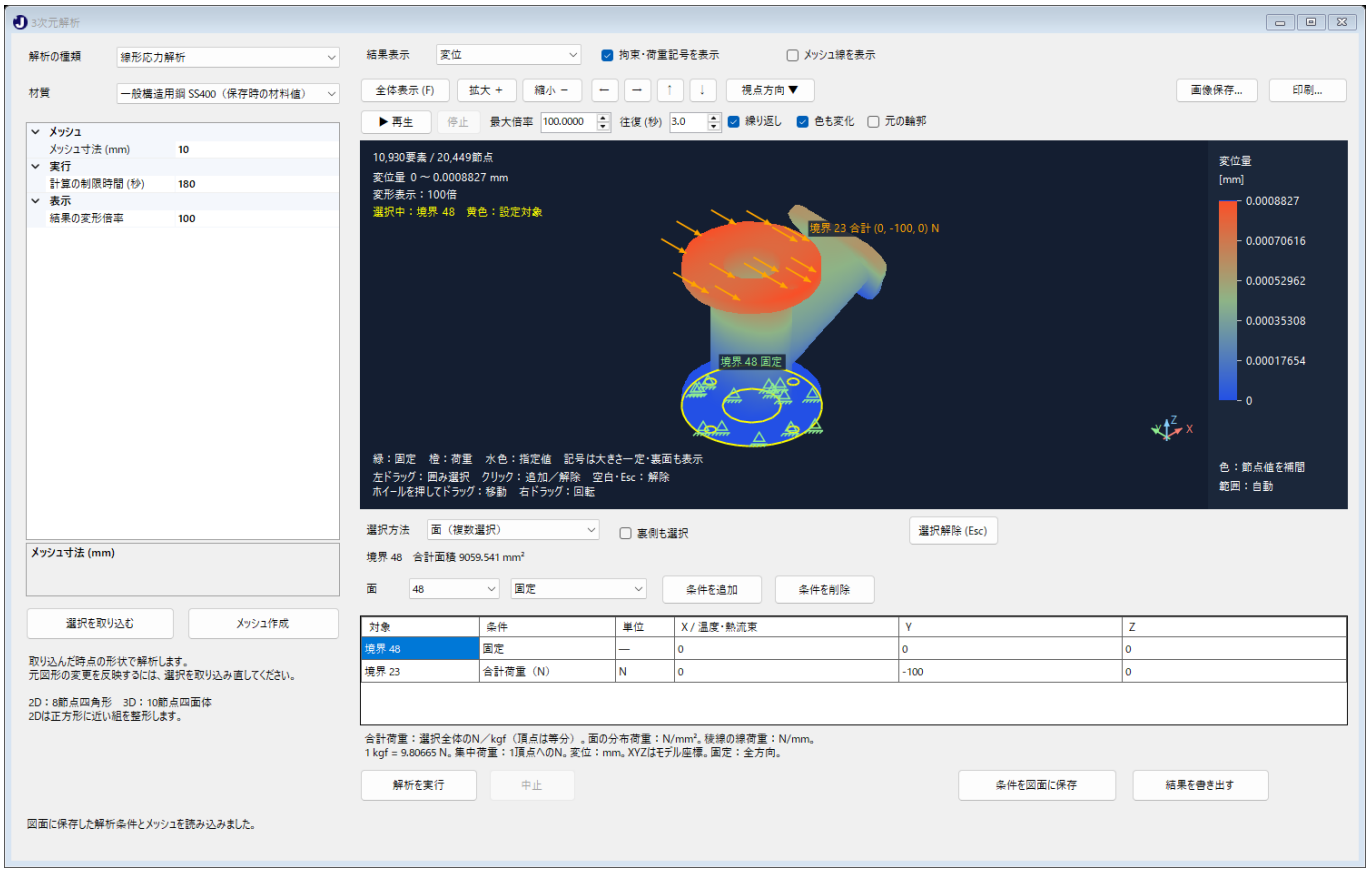


Figure 4-4. 3D displacement after meshing and setting conditions. Faces, edges, and vertices can be selected after mesh generation.

After generating the mesh, select faces, edges, or vertices and add conditions. X, Y, and Z for 3D loads and displacements use model coordinates. Check components and units in the conditions table before running the analysis.

If clicking or dragging does not select targets

Check whether the upper-left label says Imported outline (mesh not generated). Targets for conditions are selected after mesh generation. CAD selection and boundary selection in the analysis window are separate operations.

Calculation cannot proceed if insufficient restraints allow the entire body to translate or rotate. Restrain faces or points to represent the actual mounting conditions.

05 Select faces, edges, and vertices

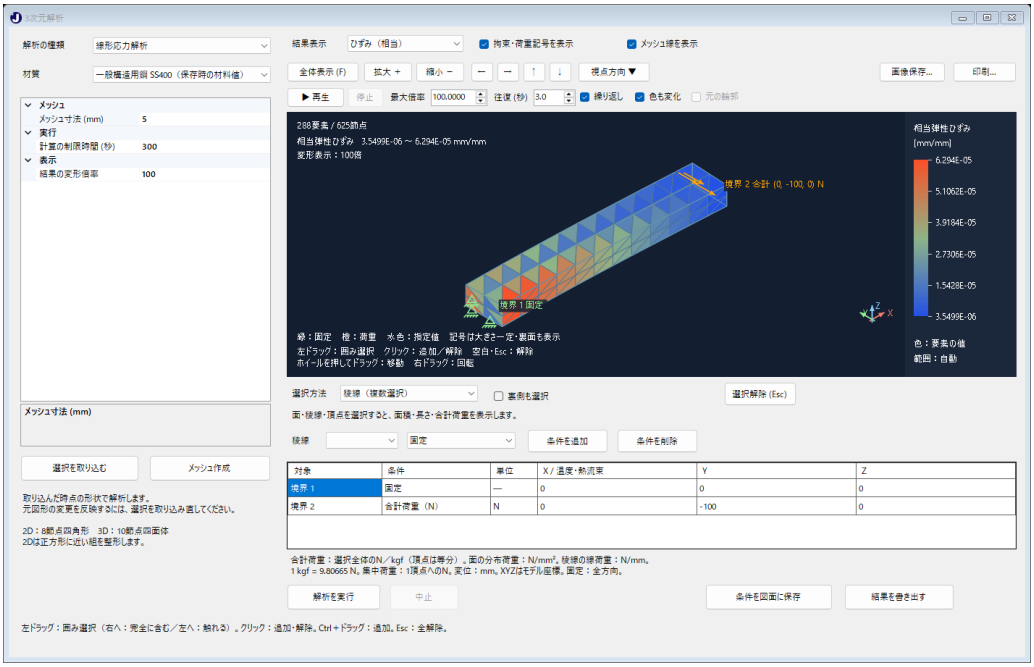


Figure 4-5. Selection mode set to Edge. Selected edges are yellow.

Operation	Selection behavior
Click	Add a target, or deselect it if already selected
Drag left to right	Select fully enclosed targets
Drag right to left	Also include targets touching the window
Ctrl + drag	Add while retaining the current selection
Select back-side targets	Include targets behind the visible surface in window selection
Click empty space or press Esc	Clear only the highlight; retain conditions and results

Use Selection mode to switch among faces, edges, and vertices and select several of the same type. Edges are CAD geometry edges, not diagonals of surface mesh triangles. The display reports face area, edge length, or vertex count.

Vertex numbers correspond to mesh node numbers. This is not direct selection of arbitrary internal 3D nodes or a portion of a face.

06 Partial selection and local meshing in 2D

In 2D, choose Entire edge, Edge range (2 points), or Node under Selection mode. Select a node for a point load.

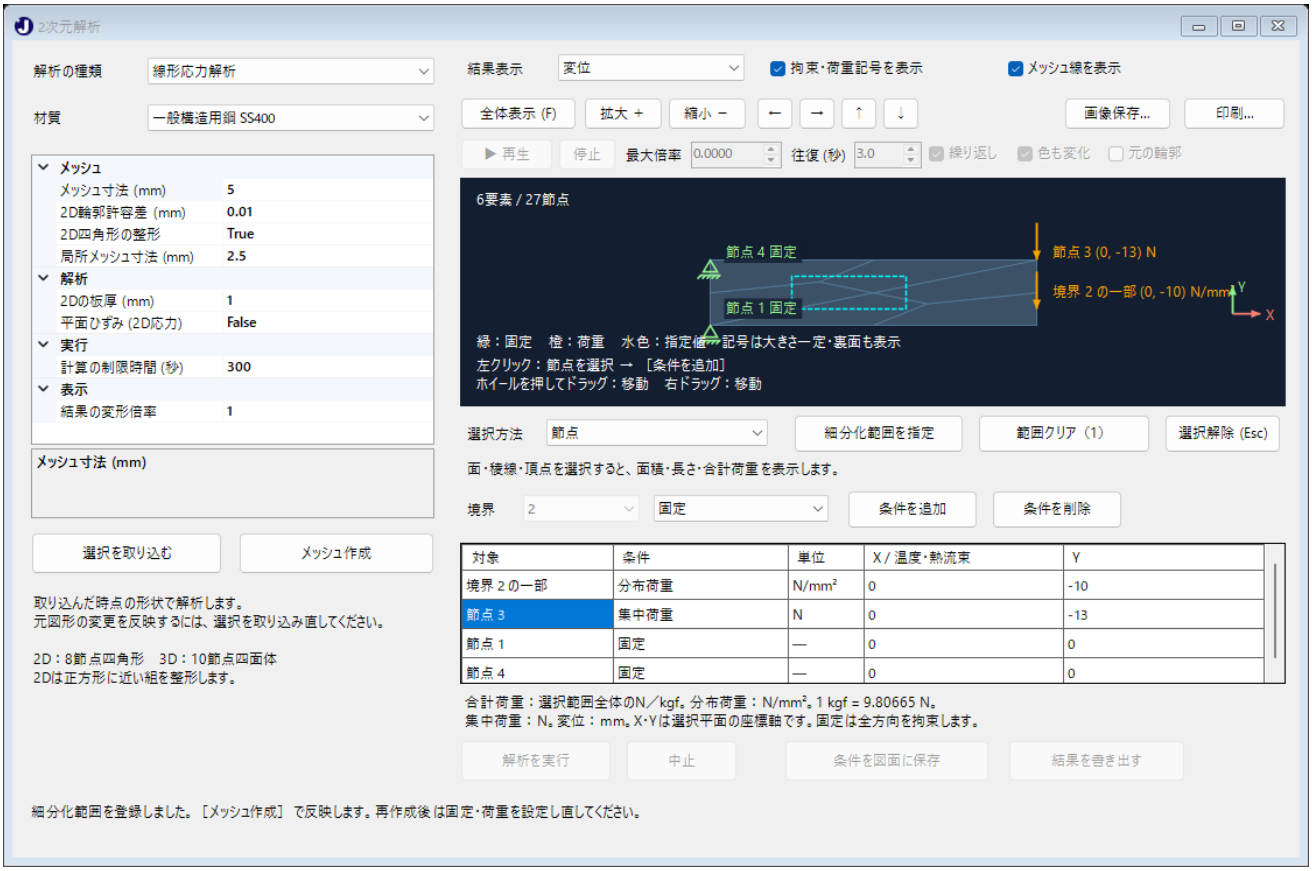


Figure 4-6. Defining a local refinement region. Regenerate the mesh after specifying the region.

Select part of an edge

With Edge range (2 points), click two points on the same boundary edge. The selected interval follows mesh subdivisions, so use a finer mesh for a smaller interval. For a closed edge, the shorter interval is selected.

Refine the mesh locally

Set Local mesh size and use Specify Refinement Region to draw a rectangle. Up to 32 regions can be registered and are applied when Generate Mesh is run. Clear Regions also takes effect on the next mesh generation.

Regenerating the mesh changes node and boundary-segment numbers, so restraints and loads must be set again. Local refinement in 3D is not supported.

07 Define restraints and loads

Target	Main stress-analysis conditions
2D edge or interval	Fixed, prescribed displacement, total load, distributed load
2D node	Fixed, prescribed displacement, point load
3D face	Fixed, prescribed displacement, total load, distributed load
3D edge	Fixed, prescribed displacement, total load, line load
3D vertex	Fixed, prescribed displacement, total load, point load

Select targets, add a condition, and enter X, Y, and Z values in the table. Fixed sets all translational components to zero; prescribed displacement specifies all translational components. These are not independent free/fixed switches for each continuum axis.

Total load and distributed load

Total load is the force over the entire selected range. Specifying 100 N on two faces does not apply 100 N to each. It is distributed by area for faces, length for edges, and count for multiple vertices.

Distributed loads use N/mm², line loads N/mm, and point loads N at one point. Total load can use N or kgf; 1 kgf = 9.80665 N. In 2D, the loaded area is edge length × thickness.

Input Example

For a total force of 100 N in negative Y, choose Total Load and set X = 0, Y = −100, and Z = 0 in 3D. This is a fixed-direction force, not a pressure automatically normal to each location on a curved face.

Loads at the same location are added. Delete the old row when replacing a condition type. Load arrows have a fixed length for readability; their lengths do not represent force magnitudes.

08 Result quantities and units

After calculation, choose a quantity under Result Display. Changing the display without changing materials, geometry, or boundary conditions does not rerun the calculation.

Result	Meaning and unit
Displacement	Displacement vector magnitude; mm
Von Mises stress	Equivalent stress; N/mm² = MPa
Reaction force	Force exerted on the body by fixed or prescribed-displacement restraints; N
Equivalent strain	Equivalent elastic strain; mm/mm
Strain X, Y, Z	Directional extension/contraction; mm/mm
Shear strain XY, YZ, XZ	Engineering shear strain; XY in 2D
Temperature	℃
Heat flux	W/m²

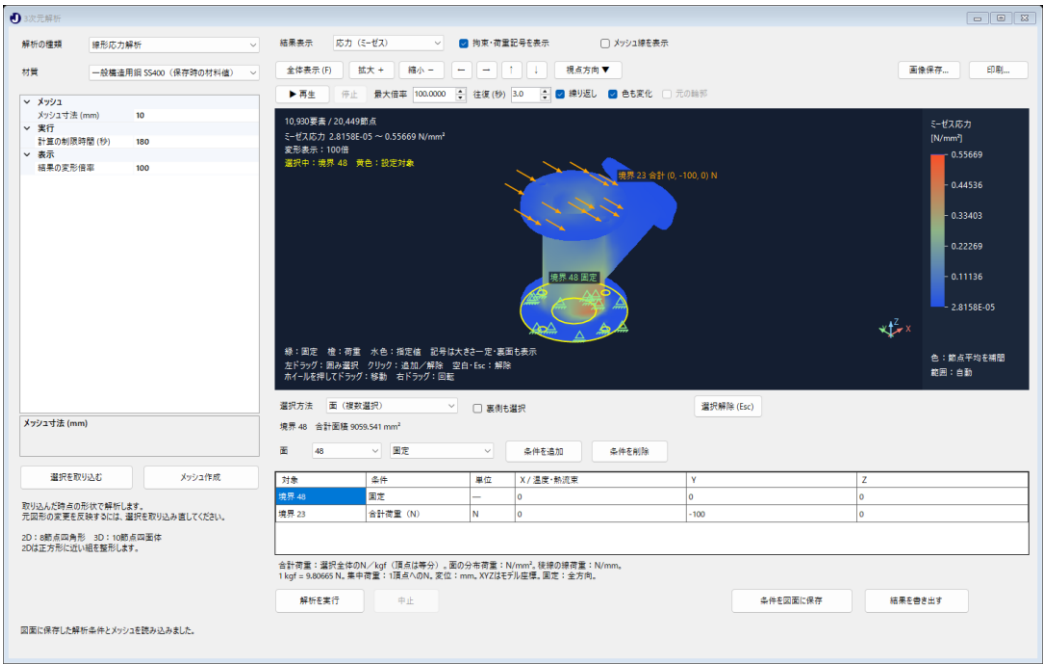


Figure 4-7. 3D von Mises stress. Check the legend's units and color range.

Deformation scale affects only the displayed shape. It does not multiply calculated displacement, stress, strain, or reaction values. The legend range is based on displayed data and may differ from extrema over all elements, including the interior.

09 Animate deformation



Figure 4-8. 2D deformation animation controls. Display settings adjust shape and color changes.

1. Run linear stress analysis and display Displacement, Von Mises Stress, or strain.

2. Set Maximum Scale. A value of 1 displays the calculated displacement; larger values exaggerate the shape change.

3. Set Round trip (s) for one animation cycle and Repeat for continuous playback.

4. Click Play. The display cycles from the original shape to the deformed shape and back.

With Animate colors enabled, colors change from the zero-value color to the result color at the same fraction as deformation. Otherwise, result colors remain fixed. Pause holds the current frame; Stop returns to the static result at maximum scale.

This display interpolates a solved result; it is not a time-dependent load or vibration calculation. Reactions are displayed statically. Disable mesh lines if a large model renders slowly. Saving a video file is not supported.

10 Overlay the undeformed outline

Set Result Display to Displacement and enable Original Outline to overlay the undeformed outline as dashed lines. Use this to compare the original position with the displayed deformation.

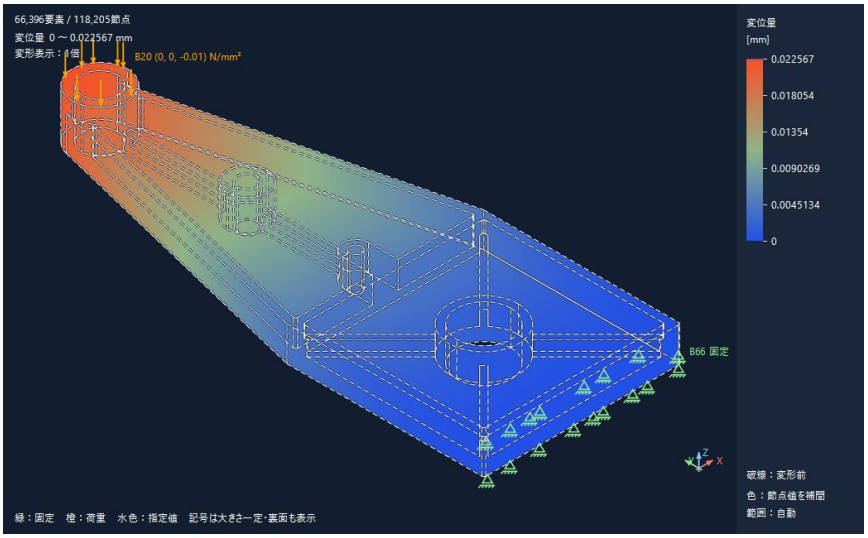


Figure 4-9. Undeformed dashed outline overlaid on the deformed geometry.

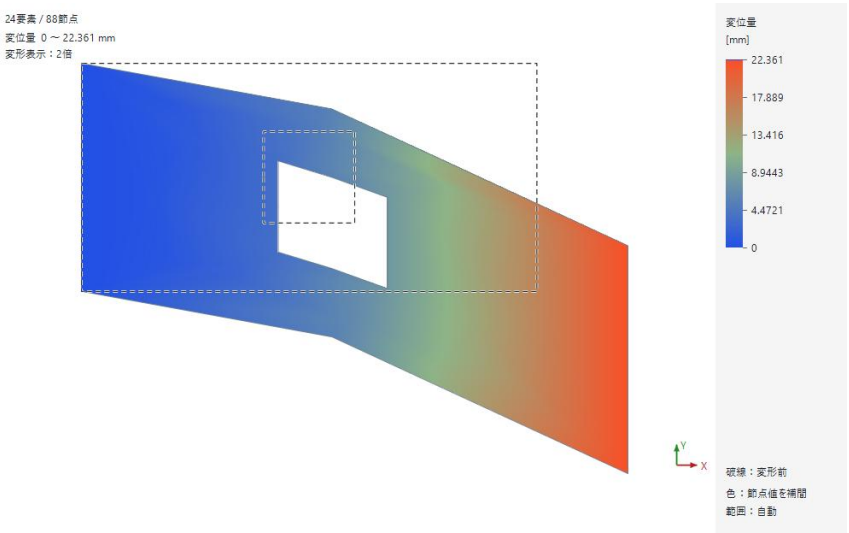


Figure 4-10. 2D outline comparison exported on a white background. Dashed lines also appear in printouts.

In 2D, the overlay shows the outer boundary and holes; in 3D, it shows face boundaries and the view-dependent silhouette. It uses the undeformed analysis mesh as its reference. Back-side lines are also overlaid in 3D.

Use Zoom Extents to fit both shapes. The undeformed outline does not move with scaling or animation. It is included in image and print outputs. This overlay is not used for stress, reactions, strain, or thermal results.

11 Mesh lines and smooth color display

Disable Show Mesh Lines to hide the grid and display a smooth distribution interpolated between nodes.



Figure 4-11. Distribution displayed with element-based colors.

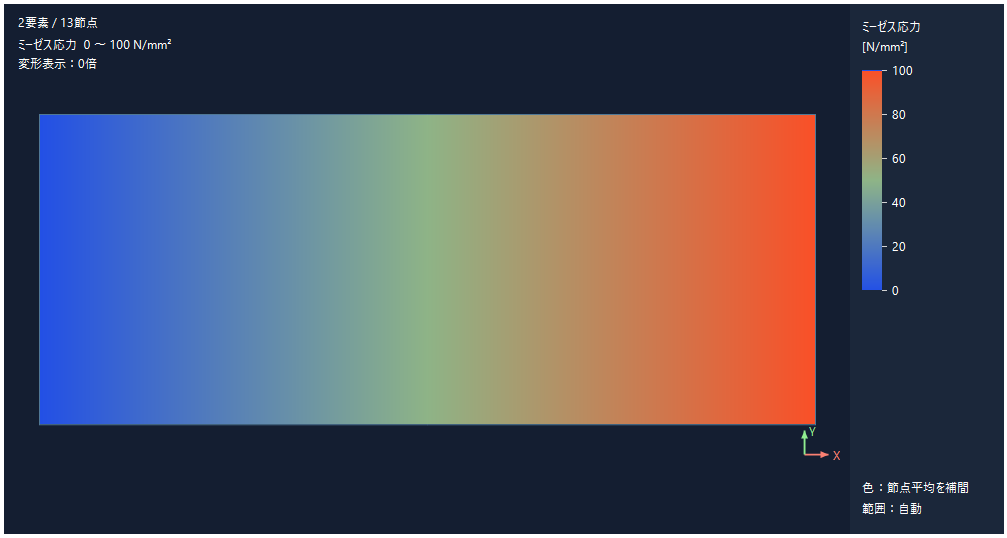


Figure 4-12. The same result with smooth colors. The display method does not change calculated values.

Displacement and temperature interpolate nodal values. Stress, strain, and heat flux average the display values of elements sharing a node before interpolation. The legend identifies the display method.

Restraint/load symbols and yellow selection highlights have separate settings and remain visible. To see only results, clear the selection and, if needed, disable Show Restraint/Load Symbols.

12 Check reactions

Choose Result Display > Reactions to show forces exerted by restraints as purple arrows. The right side displays total Rx, Ry, and Rz over all restraints in N; 2D displays Rx and Ry.

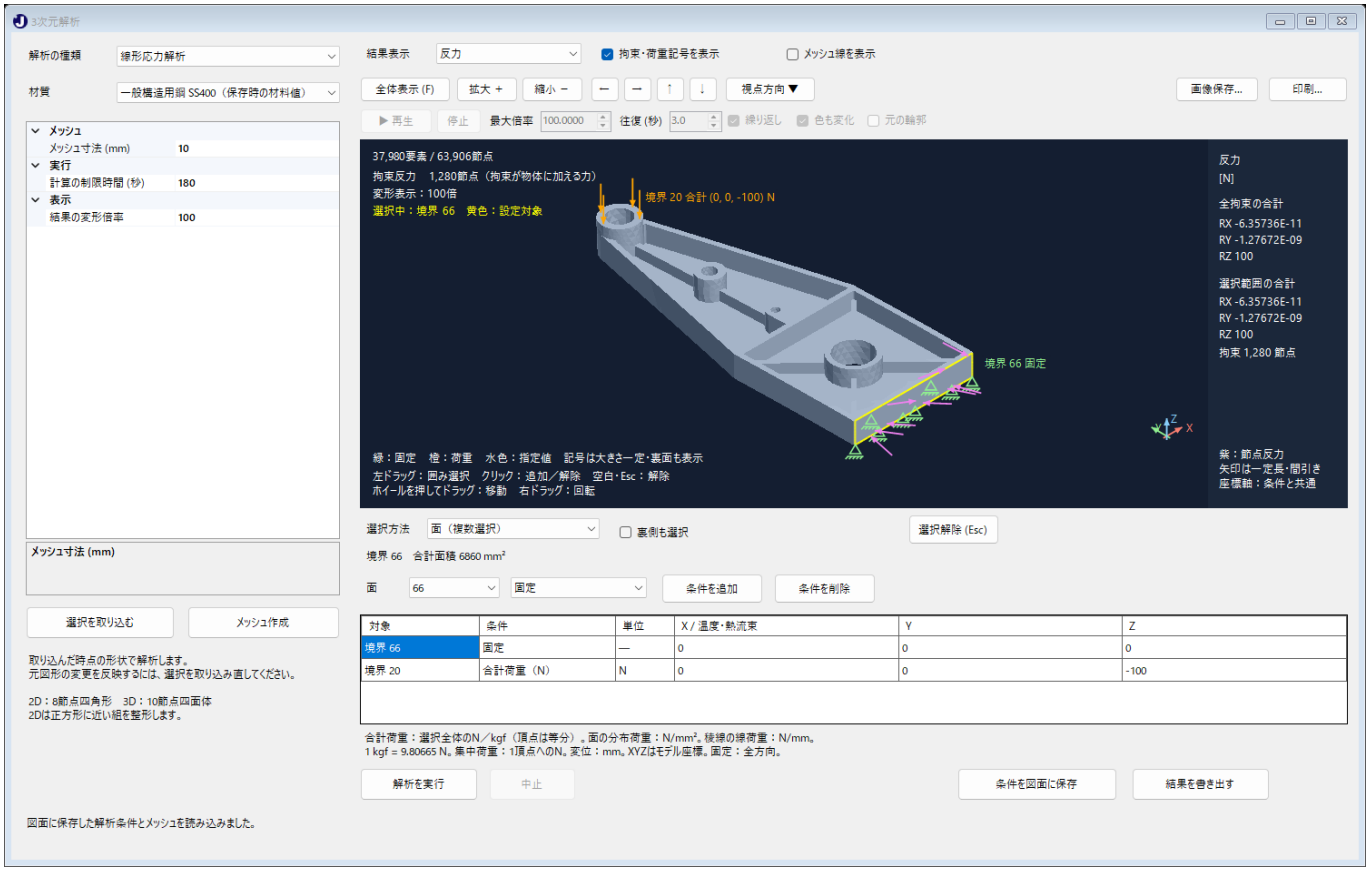


Figure 4-13. 3D reactions. Distinguish the selected-range total from the total for all restraints.

Selecting faces, edges, or nodes also displays the summed reactions at restrained nodes in that range. A range without restraints gives zero. Directions follow the condition-input coordinates: the analysis plane in 2D and model coordinates in 3D.

Arrows indicate direction with fixed length and are thinned out in dense regions. Totals and CSV output use all values and do not count shared nodes twice.

In Export Results, choose Nodal Reactions CSV to save undeformed coordinates, directional reaction components, and reaction magnitudes. Reaction displays are also included in saved images and printouts.

13 Check strain

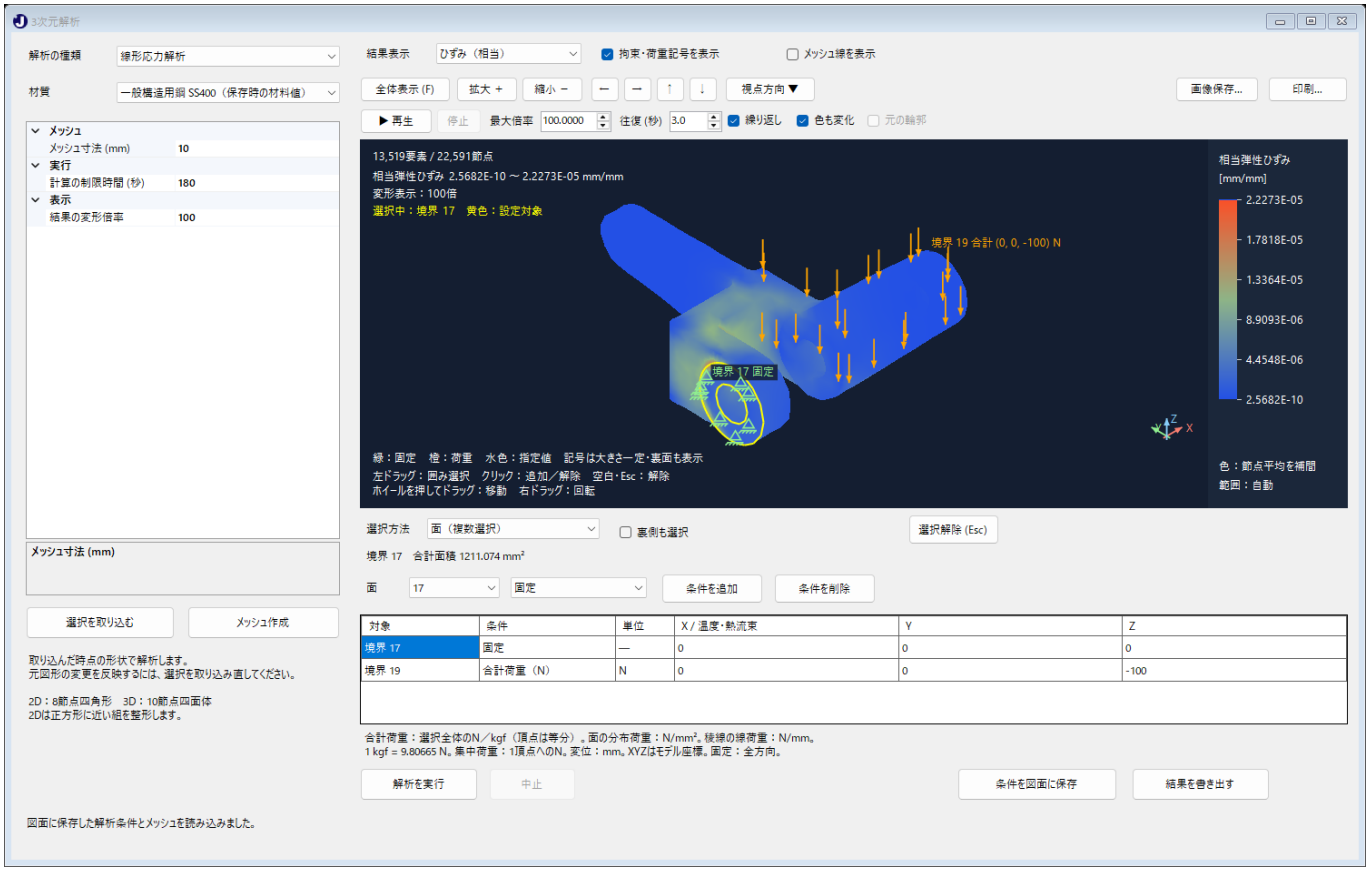


Figure 4-14. Strain distribution. Check the legend's component and signs.

Equivalent Strain is nonnegative equivalent elastic strain. Under the current isotropic linear-elastic definition, it equals von Mises stress divided by Young's modulus. It is not plastic strain.

Strain X/Y/Z are normal strains: positive for extension and negative for contraction. In 2D, Z is the thickness direction. Under plane stress it reflects Poisson-related thickness change; under plane strain it is zero.

Shear Strain is engineering shear strain γ, twice the off-diagonal strain-tensor component. Choose XY in 2D, or XY, YZ, and XZ in 3D.

Units are mm/mm; 0.001 equals 0.1%. Element Strain CSV saves each component and equivalent elastic strain. Use CSV to inspect values including internal 3D elements.

14 Run steady-state thermal analysis

Steady-state thermal analysis finds the condition after enough time has elapsed for temperatures to stop changing. Geometry import and meshing follow the same process as stress analysis.

1. Select Steady-State Thermal Analysis and check the material's thermal conductivity.

2. Select faces, edges, or other targets and add prescribed temperature, heat flux, or convection conditions.

3. Run Analysis and inspect Temperature and Heat Flux.

Thermal condition	Input and effect
Prescribed temperature	°C; maintain the specified temperature at the target
Heat flux	W/m²; positive inward, negative outward
Convection	Heat transfer coefficient in W/(m² K) and ambient temperature in °C
Insulation	No numeric value; exclude the face or edge from bulk convection

Boundaries without a condition are insulated by default. In steady-state analysis, each independent region needs a prescribed temperature or convection condition. If everything is insulated and only one point is held at 100°C, the entire body eventually reaches 100°C. The surroundings do not automatically become 20°C.

To test a temperature distribution, set a rectangle's left edge to 100°C and right edge to 20°C, leaving the rest insulated. With constant thermal conductivity, a temperature gradient develops between these edges.

In 3D, faces support prescribed temperature, heat flux, convection, and insulation. Edges and vertices support prescribed temperature.

15 Run transient thermal analysis

Transient thermal analysis calculates temperature changes from an initial temperature over time. It uses density and specific heat as well as thermal conductivity.

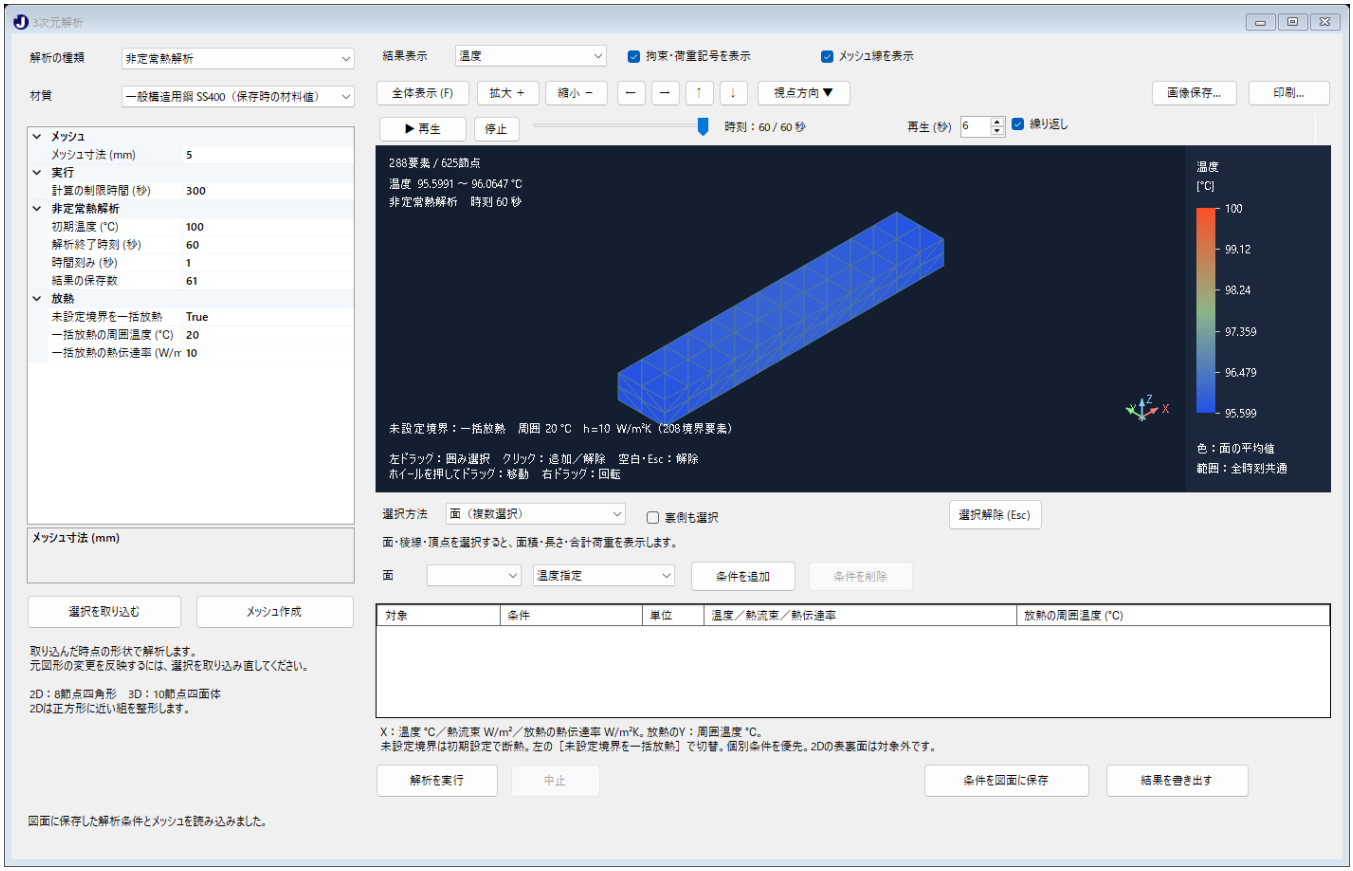


Figure 4-15. Transient thermal analysis. Time and convection settings are on the left, with a time slider at the top.

1. Set Analysis type to Transient Thermal Analysis.

2. Set Initial temperature, End time, and Time step.

3. Define temperature, heat flux, convection, or other conditions and run the analysis.

4. Select a time with the slider or click Play to observe temperature changes.

For example, an initial temperature of 100°C, end time of 60 s, and time step of 1 s calculate the process from 100°C through 60 seconds. Calculation time limit is the maximum processing time allowed, not the physical elapsed time.

Prescribed Temperature holds its target at the same temperature throughout the analysis. To observe natural cooling only, set an initial temperature and convection without a prescribed-temperature condition.

16 Apply convection to all unset boundaries

Under Convection on the left, set Apply convection to unset boundaries to True, then enter the ambient temperature and heat transfer coefficient. This is available for both steady-state and transient analysis.

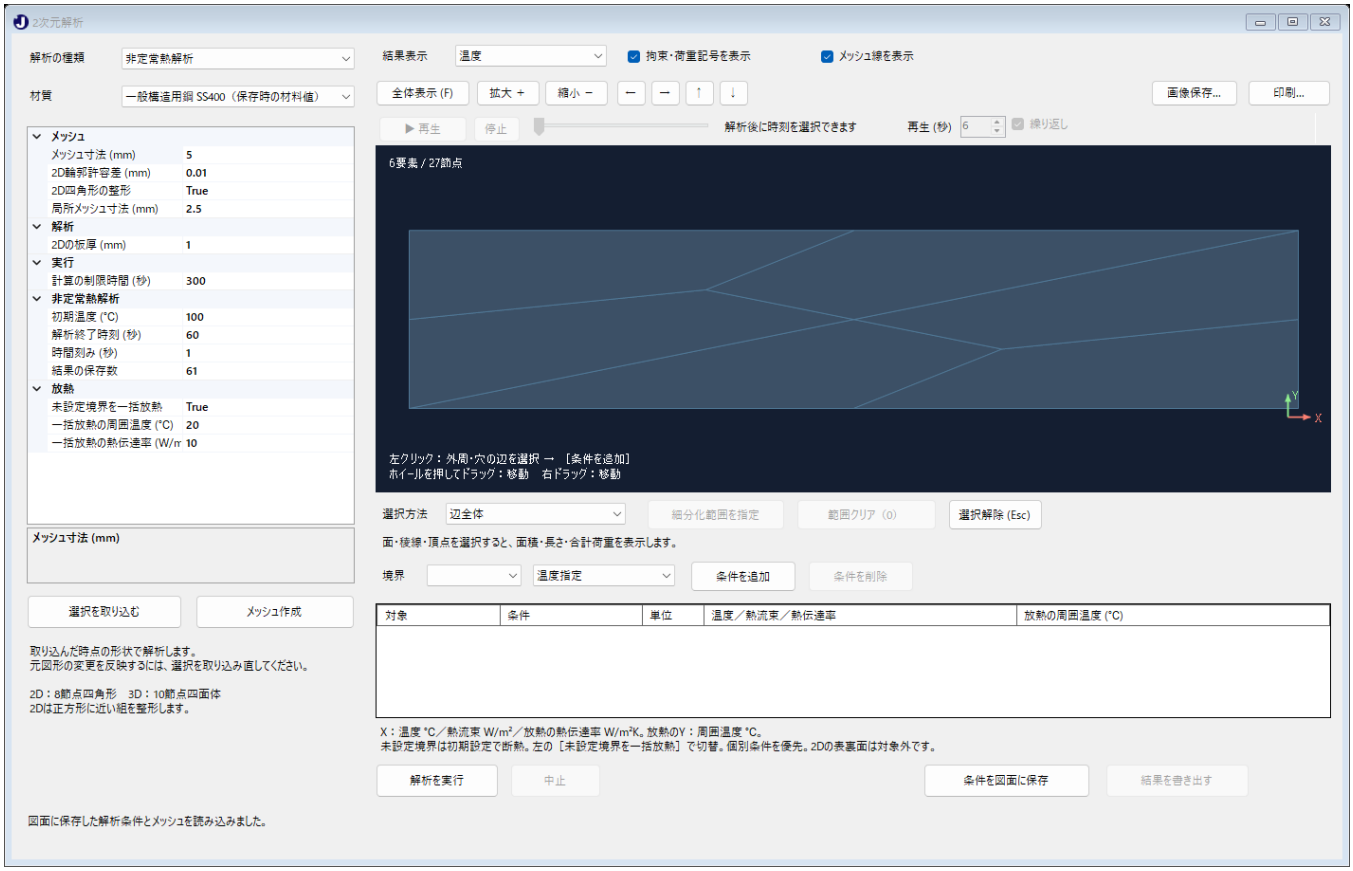


Figure 4-16. 2D example with bulk convection for unset boundaries enabled.

Where bulk convection applies

In 3D, it applies to external surfaces. In 2D, it applies to outline edge length × thickness. It does not include convection from the broad front and back faces of a 2D plate.

Faces or edges with individual temperature, heat flux, convection, or insulation conditions are excluded. For a partial 2D edge condition, only that interval is excluded. Temperature conditions on edges, vertices, or nodes do not exclude the entire adjacent face or edge from convection.

The initial heat transfer coefficient of 10 W/(m² K) is a placeholder. Adjust it for the actual fluid, velocity, and other conditions. If the ambient temperature exceeds the component temperature, convection heats the component.

17 Play and save time-step results

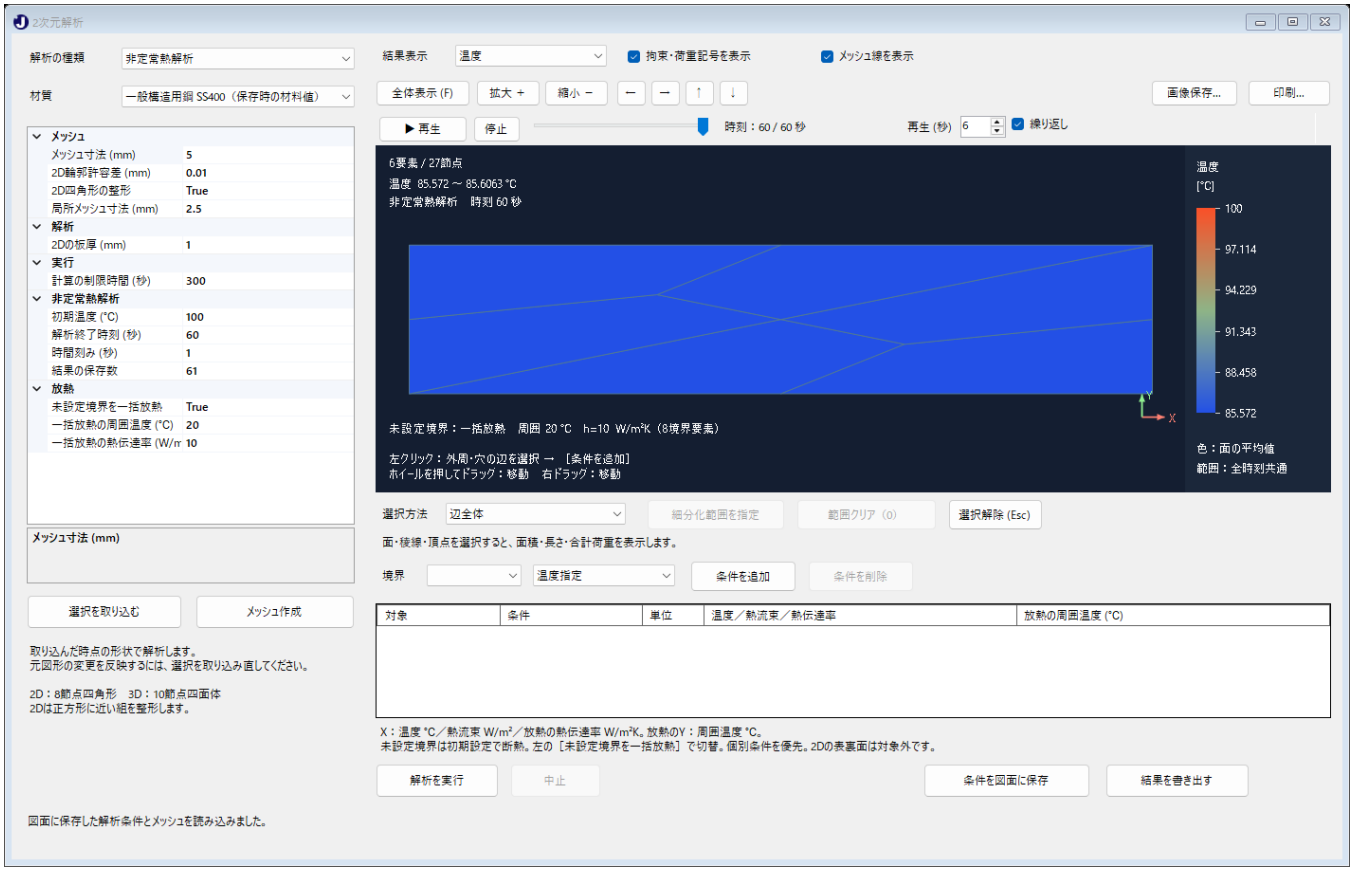


Figure 4-17. Temperature at the selected time. A common legend is used for all saved times.

Use the slider and Play, Pause, and Stop controls. Stop in thermal analysis returns to the starting time. Playback (s) controls display duration and does not change the calculated physical time.

Result storage count is the approximate number of saved times, including the start and end. Reducing it does not change the calculation time step. For a large mesh, reduce the count to stay within memory limits.

Heat Flux is also displayed from the temperature at the selected time. Pause at the required time before saving an image or printing; the time and convection settings are included. JSON can store all saved times, and CSV can export nodal temperatures by time.

Scope

Material properties and boundary conditions remain constant during the analysis. Time-dependent temperature/heat-flux tables, radiation, phase change, temperature-dependent properties, fluid analysis, and coupled thermal stress are not included. Compare with a smaller time step to assess its influence.

18 Run beam calculations

Open Analysis > 2D Analysis or 3D Analysis > Beam Calculation. This handles straight beams with uniform material and cross-section over their entire length.

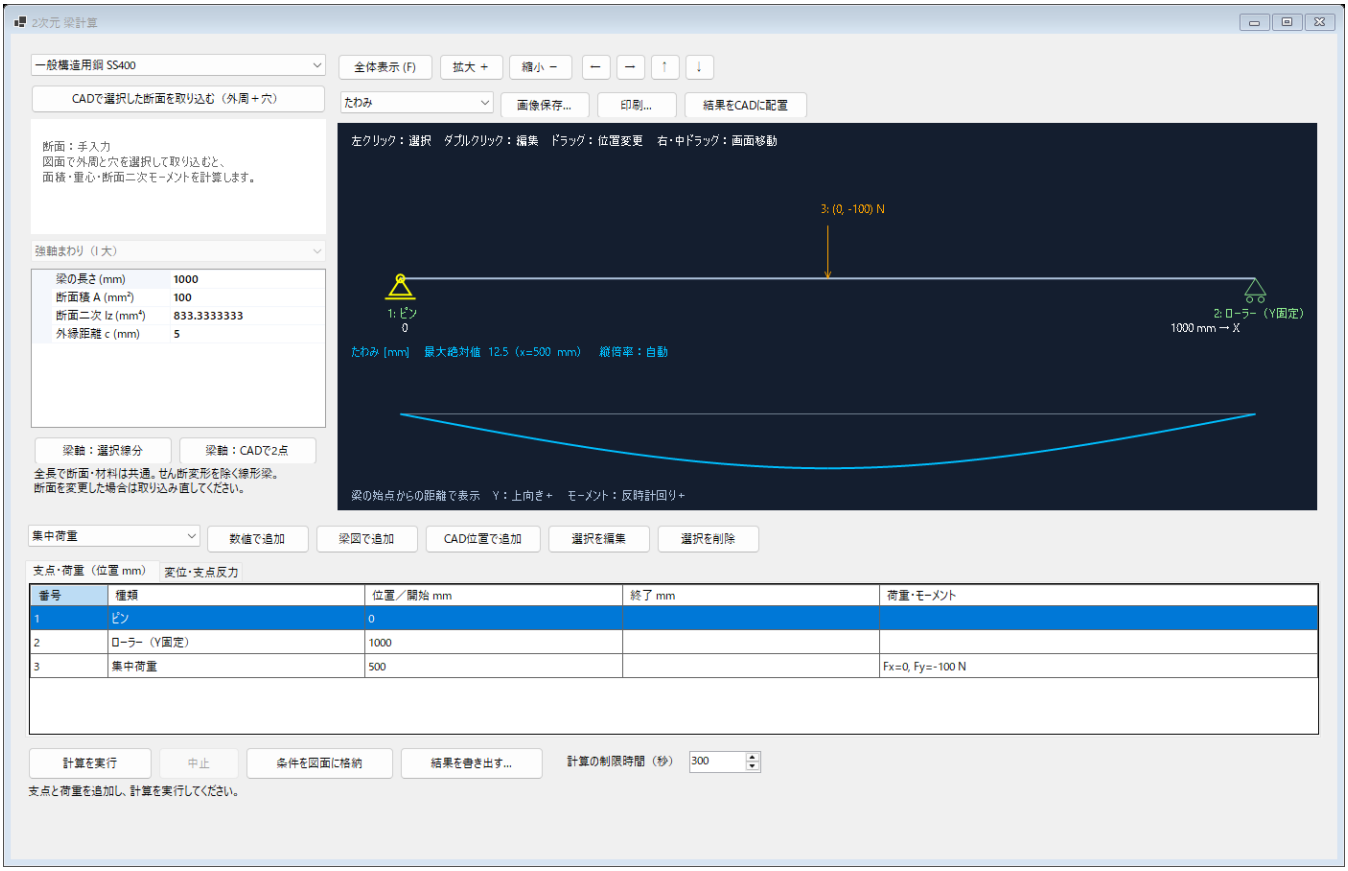


Figure 4-18. 2D beam calculation, showing the section, supports, loads, and results.

1. Use Import Selected CAD Section to import a planar face or outer and hole boundaries. If nothing is selected, select in CAD and press Enter.

2. Import the beam axis from a line or two CAD points. The length can also be entered directly.

3. Choose a support or load type and add it by value, on the beam diagram, or at a CAD location. For distributed loads, specify start and end points.

4. Run Calculation and switch among deflection, shear force, bending moment, normal stress, and other results.

In 2D, upward Y is positive, so downward loads are negative. Point loads use N, distributed loads N/mm, and moments N·mm. Double-click the table or beam diagram to edit supports and loads.

For manual section input, enter area A, second moment of area I, and extreme-fiber distance c. Bending stress is not evaluated when c = 0.

19 3D beam coordinates and sections

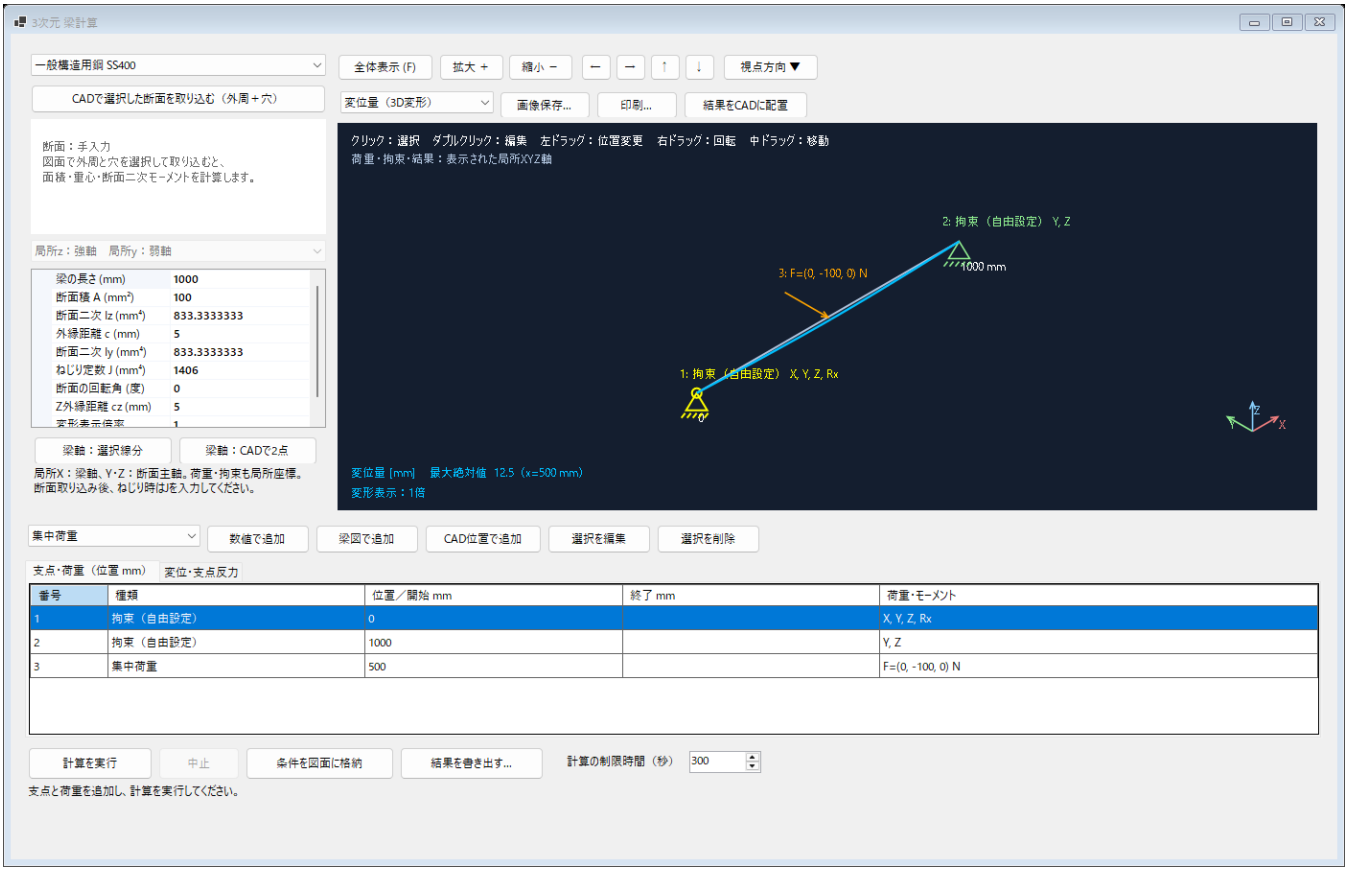


Figure 4-19. 3D beam calculation, showing the section, supports, loads, and results.

For a 3D beam, X runs from the beam start to its end; Y and Z are the section's principal axes. Enter loads and restraints in the displayed local coordinates. Use Section rotation to adjust orientation about the beam axis.

Set each of the six restraint directions as free or fixed. A restraint against rigid-body torsional rotation is also required. Positive moments follow the right-hand rule about each axis.

CAD sections provide area, centroid, second moments of area, and related properties, but the torsion constant J is not estimated automatically. If applying Mx, enter a J appropriate to the section. Do not substitute Iy + Iz for J.

Both 2D and 3D beams use small-deformation linear elasticity. Shear deformation, plasticity, buckling, large deformation, and varying cross-sections are excluded. Normal stress combines axial and bending effects; it is not von Mises stress.

Place Results in CAD inserts graphs or other results into the drawing. In 3D displacement-magnitude mode it inserts a deformed curve; other modes insert a result graph on the work plane.

20 Trusses and frames

Select lines or straight polylines in CAD and import them into Truss or Frame analysis. Trusses use pin-jointed members carrying axial forces; frames use rigid joints that transmit bending.

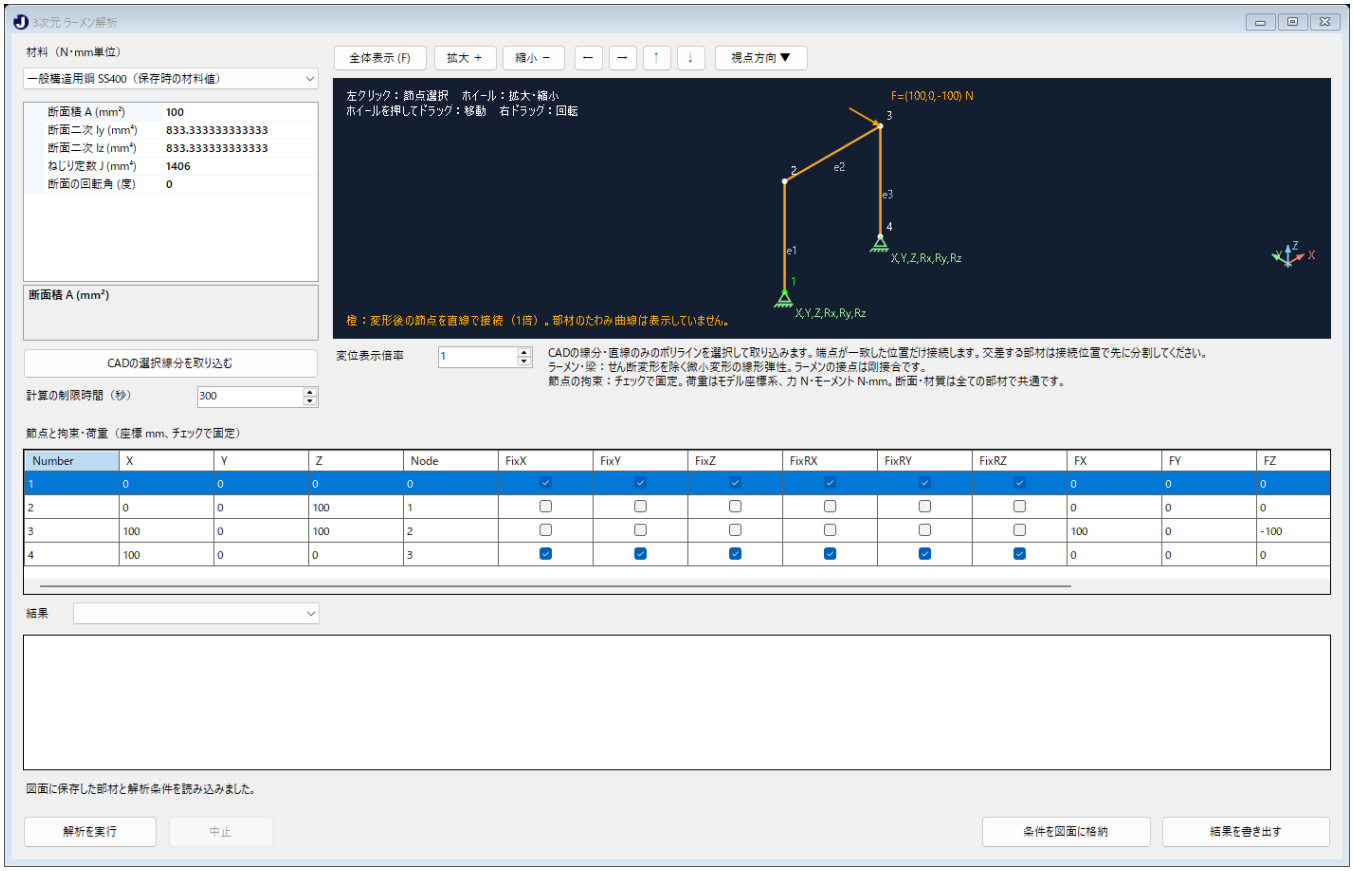


Figure 4-20. Nodes and conditions in a 3D frame. Checked degrees of freedom are restrained.

1. Import members and set the material and common section properties.

2. Click a node to select its table row, then enter restrained directions and loads.

3. Run Analysis and inspect nodal displacements, reactions, and member forces.

Members connect at coincident endpoints. Lines that merely cross do not connect; split them beforehand at intended joints. In 2D, all members must lie in one plane parallel to the model XY plane.

The deformed display joins displaced nodes with straight lines; it does not show member deflection curves. All members share material and section values. The limit is 2,400 degrees of freedom.

21 Export images, printouts, and values

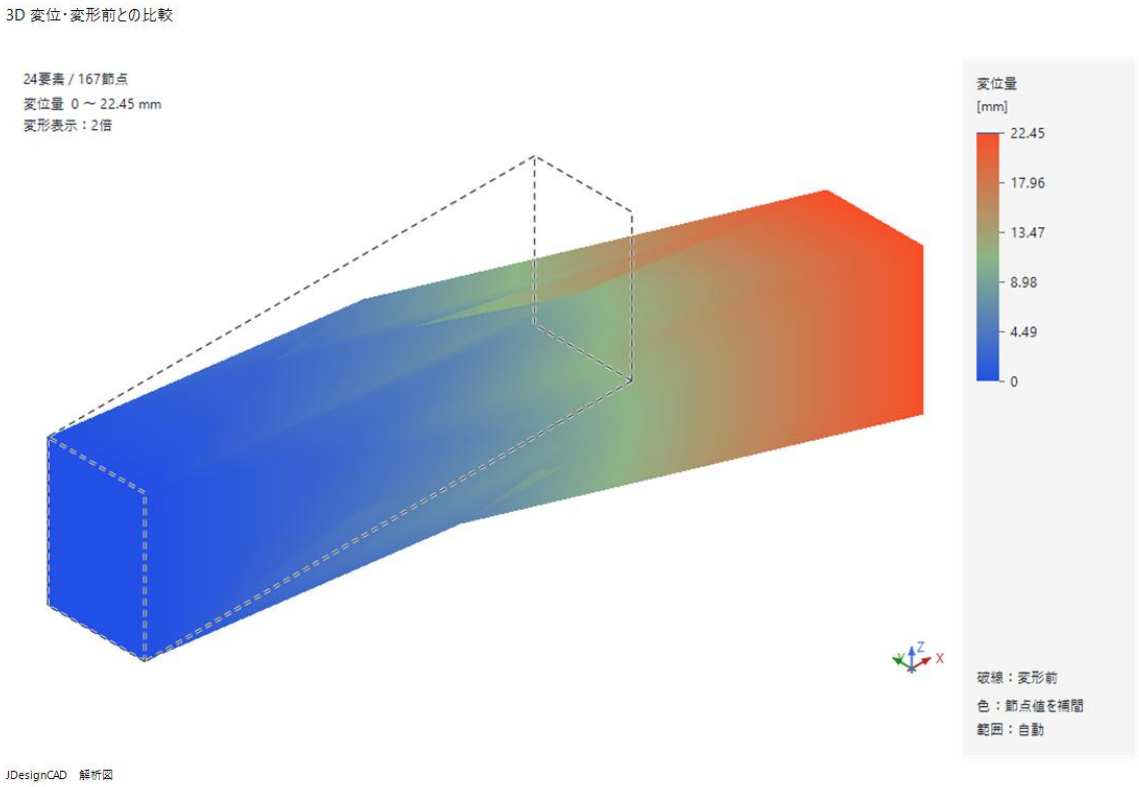


Figure 4-21. White-background print page showing geometry orientation, results, legend, deformation scale, and other information.

Save Image… saves the current display as PNG or JPEG. Print… opens a white-background preview; choose paper and printer before printing. Opening the preview alone does not print.

To include the entire model, first use Zoom Extents and configure mesh lines, symbols, and the original outline. Pause animation to export that frame. Yellow selection highlights and operating instructions are excluded from image output.

Export Results outputs numeric data. In addition to JSON, available CSV types depend on the result and include nodal reactions, element strains, and temperatures by time. CSV node and element numbers start at 1.

22 Save conditions in the drawing

In the analysis window, click Save Conditions in Drawing or Store Conditions in Drawing, then save the JDC normally. The analysis-window button alone does not save the file.

Stored data includes material values, the mesh or members, boundary conditions, and transient time/convection settings. Recalculate after reopening the drawing to obtain results. Use Export Results separately to retain numeric results.

Changed item	Required next action
CAD geometry	Reimport, regenerate the mesh, and redefine conditions
Mesh size or outline tolerance	Regenerate the mesh
Material, boundary conditions, or physical settings	Rerun the analysis
Displayed quantity, deformation scale, or viewpoint	Update the display only; no recalculation
Transient initial temperature or time step	Rerun the analysis

One condition set is stored for each 2D/3D analysis category. Steady-state and transient thermal analysis share conditions; save separate JDC files if both setups must be retained independently.

If calculation fails

First check for closed geometry, correct material values and units, insufficient restraints, and conflicting prescribed temperatures. Use Stop or the calculation time limit to halt processing. If geometry was simplified, also check that the physical problem has not changed.

Near sharp corners or loads concentrated at points or edges, maximum stress may depend strongly on mesh size. Compare with a finer mesh and do not base conclusions on the maximum value alone.

23 Example: a component with ribs and holes

This 3D analysis example uses the solid in 練習 16.jdc (Exercise 16). Its elongated shape includes several holes and reinforcing ribs, allowing examination of deformation and distributions beyond a simple beam model.

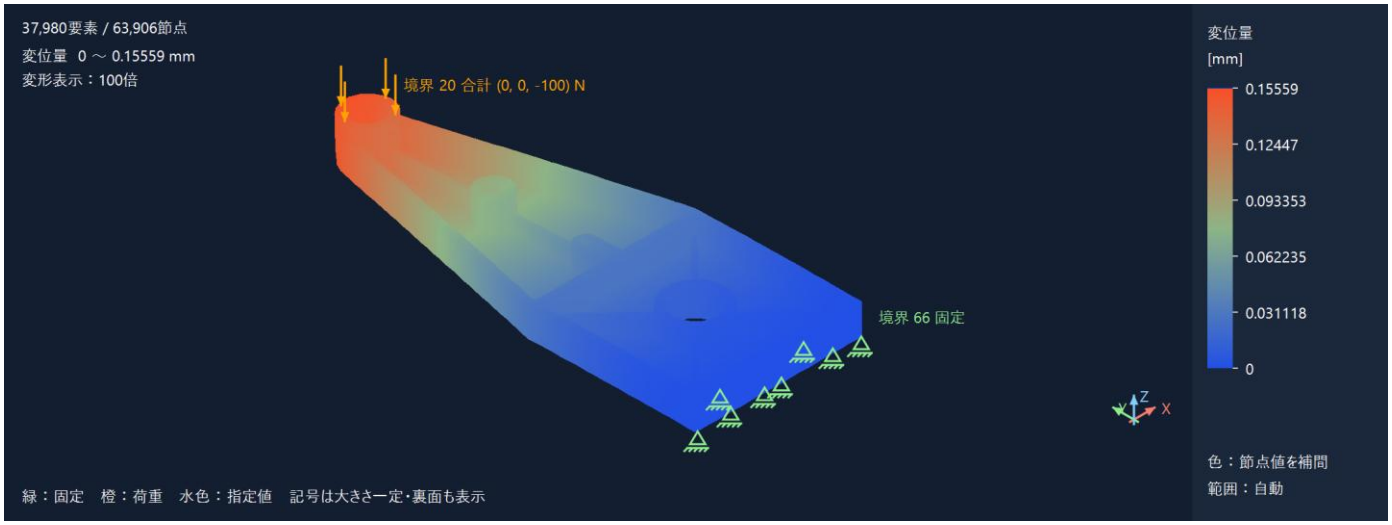


Figure 4-22. Displacement in Exercise 16, displayed at 100× deformation scale.

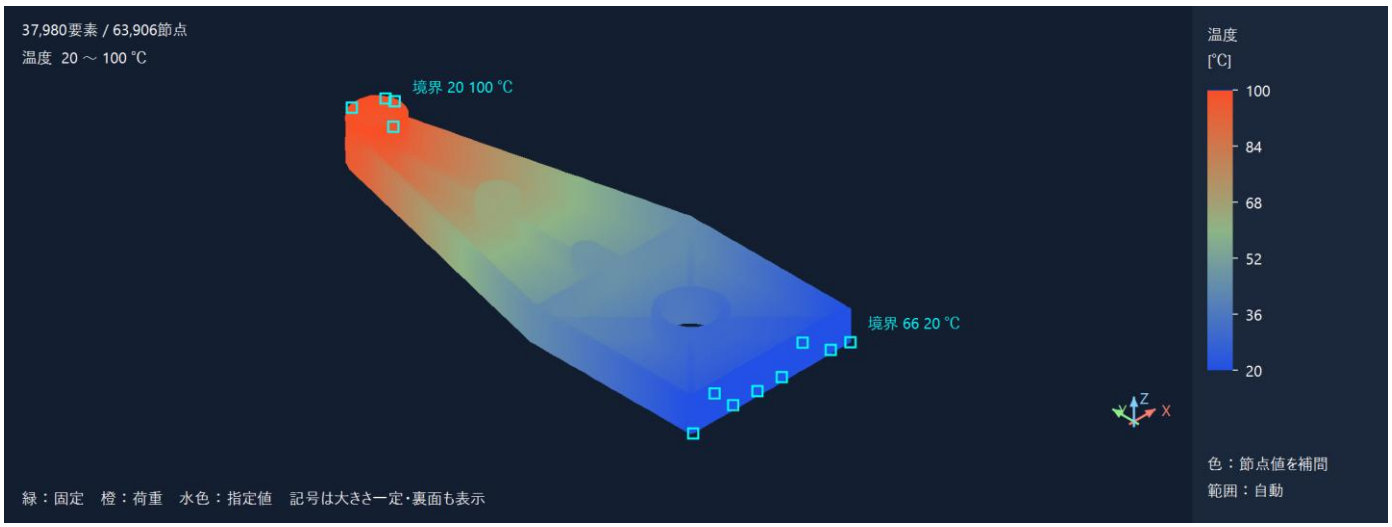


Figure 4-23. Steady-state temperature in the same geometry. A gradient forms between prescribed temperatures of 20°C and 100°C.

Mechanical settings are SS400, mesh size 10 mm, the end face at minimum Y fixed, and a total Z load of −100 N on the upper face near the tip. For thermal analysis, the previously fixed face is set to 20°C, the loaded face to 100°C, and all remaining boundaries insulated.

These conditions illustrate operation and display. Replace them with the actual mounting, loads, material, and heat-transfer conditions.

24 Example: a component with flanges and curved faces

Examples using 練習 8.jdc and 練習 13.jdc (Exercises 8 and 13) show results for geometry with flanges, holes, and curved faces. Switch result types to inspect displacement, stress, strain, and reactions from the same structural conditions.

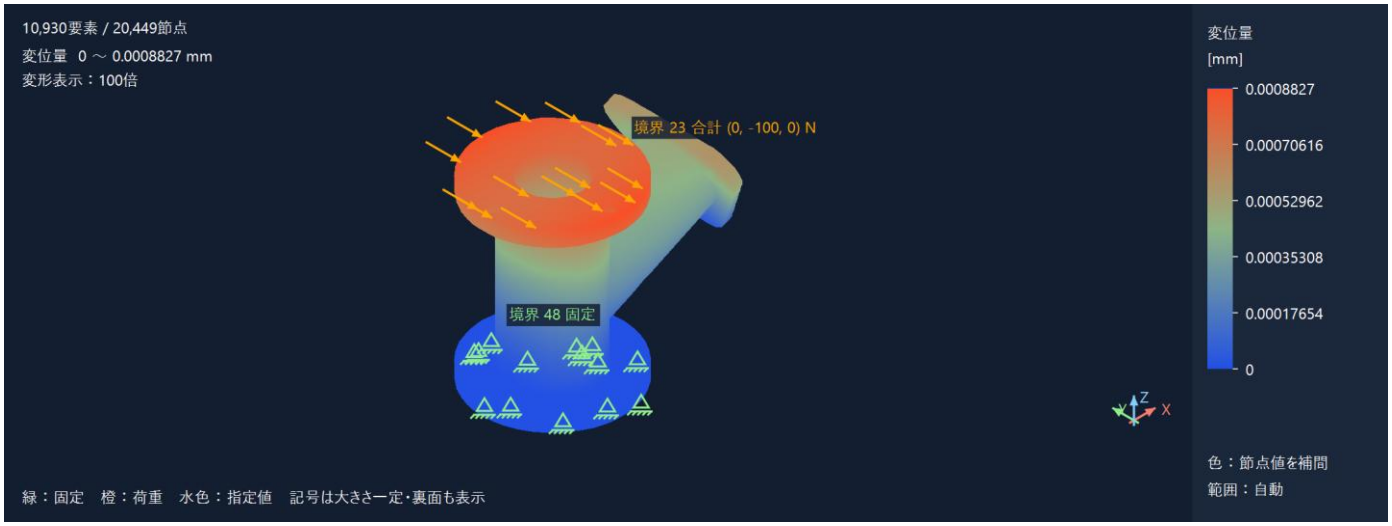


Figure 4-24. Exercise 8 displacement. The minimum-Z face is fixed, and a total Y load of −100 N is applied to the maximum-Z face.

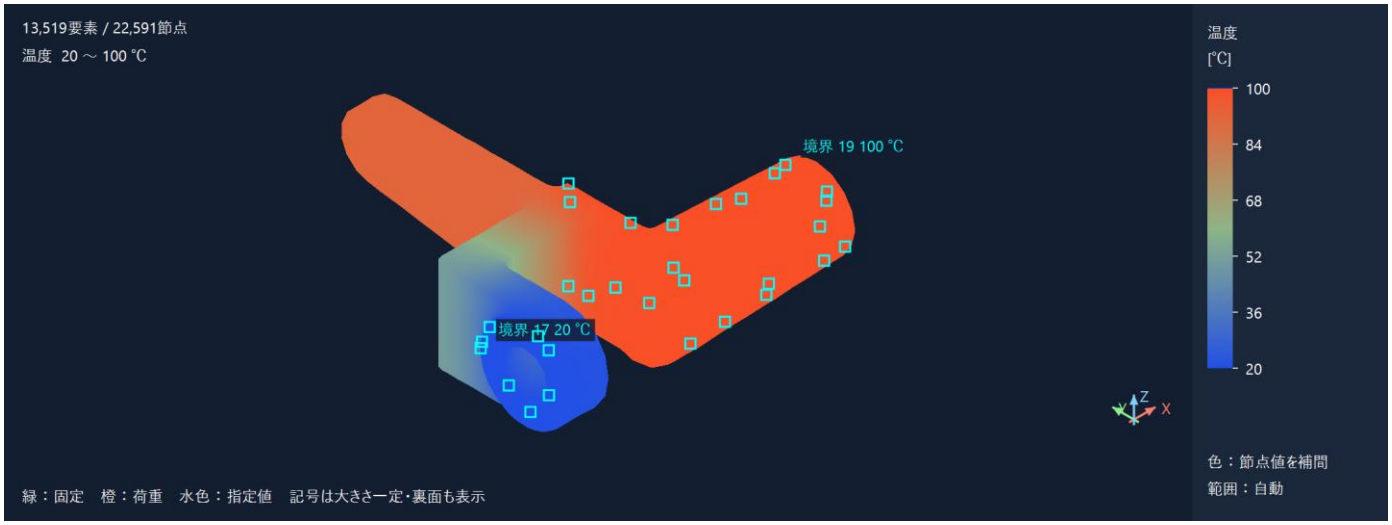


Figure 4-25. Exercise 13 steady-state temperature. The minimum-X end face is 20°C, the maximum-X end face is 100°C, and the rest is insulated.

Both examples use SS400 and a 10 mm mesh size. Figure 4-24 displays deformation at 100× scale. Retain units and color ranges in exported images; do not compare magnitudes from geometry colors alone.

For design evaluation, verify the conditions and compare with finer meshes as needed.

25 Choose the appropriate analysis

Analysis is available in STD and Pro. First decide which quantities and geometry are needed. Investigating a simple bar's deflection requires a different model from investigating stress around a hole.

Question to investigate	Function	Required input
In-plane deformation and stress of a thin plate	2D stress analysis: plane stress	Closed outline/face and thickness
A section restrained against out-of-plane deformation	2D stress analysis: plane strain	A justified plane-strain assumption
Stress in thick parts, holes, or complex geometry	3D stress analysis	Closed solids, materials, and bonding
Deflection and bending moment of a straight beam	2D/3D beam calculation	Axis, section, supports, and loads
Axial forces in a pin-jointed structure	Truss	Connected straight members
Bending and axial forces in a rigid-jointed structure	Frame	Straight members and section properties
Temperature after a sufficiently long time	Steady-state thermal analysis	Material, temperatures, heat flux, and convection
Cooling or heating over time	Transient thermal analysis	The above plus initial temperature, time, density, and specific heat

Modeling limitations

Structural analysis of faces and solids uses small-deformation linear elasticity. 2D plane stress addresses in-plane behavior of thin plates; it is not a plate element for out-of-plane bending. Beam models approximate slender straight members and do not calculate local stresses around hole edges.

Plasticity, large deformation, buckling, vibration, separating/sliding contact, coupled thermal stress, automatic self-weight loads, and design-code pass/fail checks are excluded. Deformation animation interpolates already calculated results; it is not vibration analysis.

Start with simple geometry to verify load directions and restraints, then add necessary detail. Check deformation direction and equilibrium between loads and reactions as well as maximum values.

26 Build the analysis conditions in order

Record before analysis

Item	Decision	Example
Objective	Values and locations to compare	Tip displacement, hole-edge stress, temperature after 60 s
Geometry	Details to retain or omit	Retain holes and rounds under evaluation
Material	Properties appropriate to the actual product	E, Poisson's ratio, thermal conductivity, etc.
Supports and bonding	Which region is fixed to what	Fixed mounting face, perfect bonding
External actions	Magnitude, direction, and extent	Total −100 N over an entire face
Units	Units of length, loads, and material properties	mm、N、N/mm²
Accuracy check	Compare mesh sizes/time steps	Reduce size and record the change

Common workflow

1. Select analysis geometry and click Import Selection. A copy of the CAD geometry becomes the analysis model.

2. Set the material and analysis type, thickness for 2D, and materials/bonding for multiple solids.

3. Generate the mesh and check missing geometry, holes, details, and interfaces.

4. Select boundaries, add restraints/loads or temperature/convection conditions to the table, and confirm the values.

5. Run Analysis and inspect distributions and values. If needed, refine the mesh, redefine conditions, and compare.

6. Store conditions in the drawing and save JDC. Export images and numeric results separately as needed.

Editing CAD geometry does not automatically update the imported analysis model. Reimport after geometry changes. Display scale, viewpoint, and color changes do not alter physical conditions.

27 Configure multiple materials and perfect bonding

Keep the two face-contacting solids separate. Each example body is 40 × 20 × 10 mm, giving a 200 mm² interface. Open 3D Analysis and choose Import Selection. Analysis uses a snapshot of the geometry at import time; reimport after changing the original.

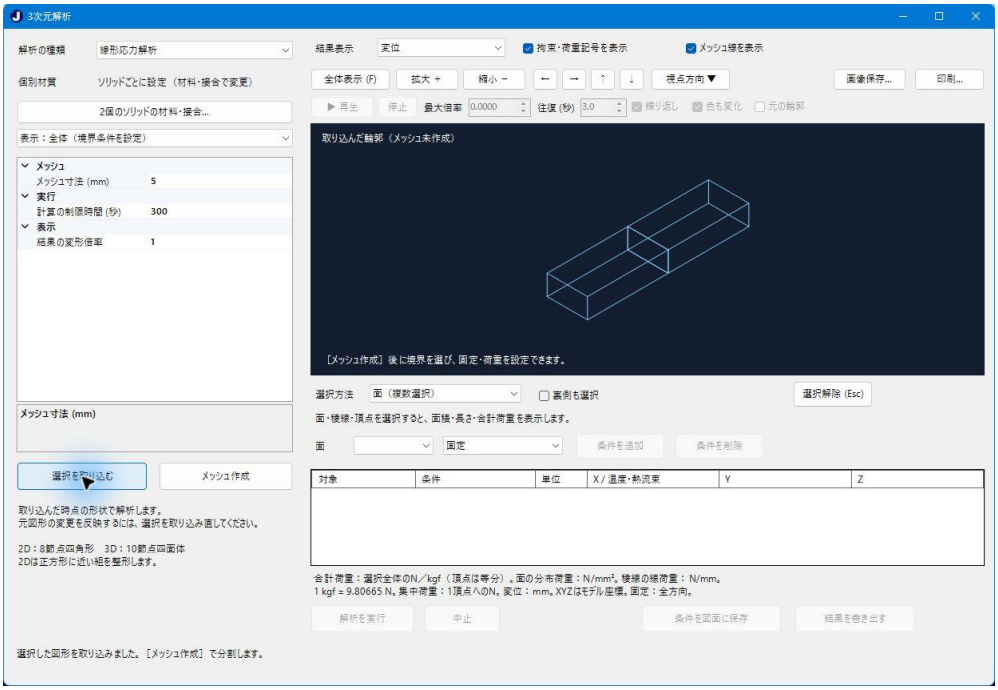


Figure 204 3D Analysis with two imported bodies

Assign a material to each body

Open Materials and Contacts for 2 Solids. On the materials tab, choose a registered material for each row. Selecting a row highlights that body in the preview. Check elastic modulus, Poisson ratio, density and units in the material catalog; the steel and aluminium shown here are examples, not validated design data.

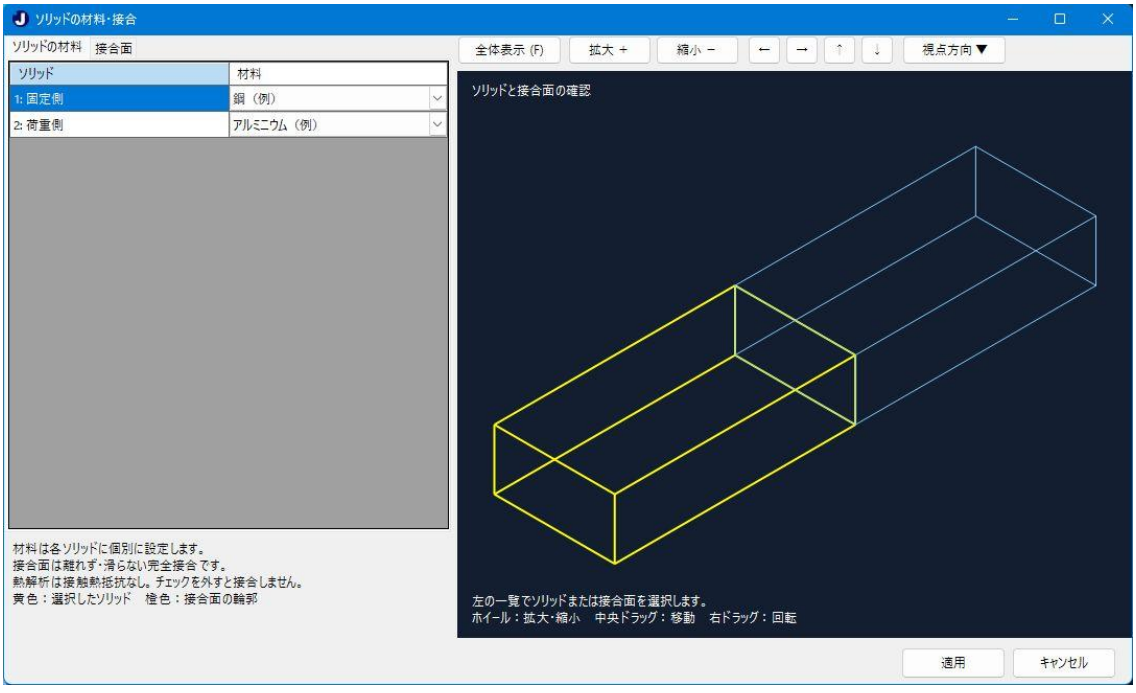


Figure 205 Material assignment and corresponding body highlight

Inspect and apply bonded contacts

On Contacts, check the body pair and area. Selecting a row displays both bodies and the interface outline. A bonded interface makes displacement and temperature continuous. This is not nonlinear contact with friction, sliding, separation or debonding. Enable the intended bonds and choose Apply.

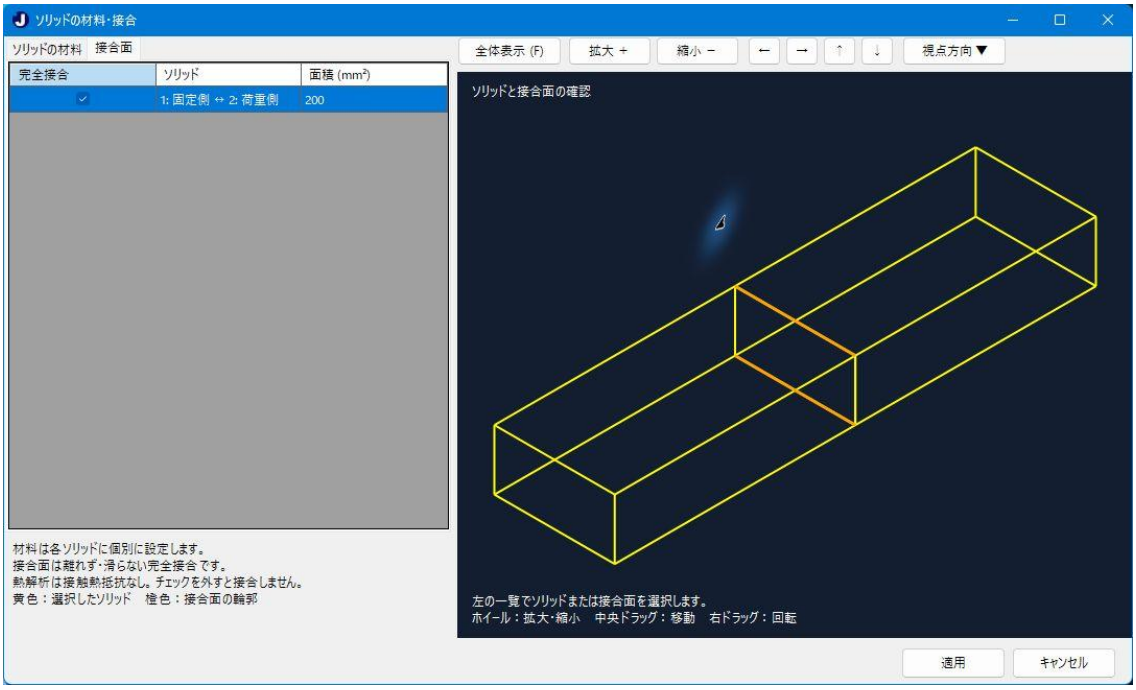


Figure 206 Inspecting the 200 mm² interface

Create the mesh, choose Entire Assembly (set boundary conditions), and assign supports and loads. Rerun analysis after changing a material. Changing which interfaces are bonded requires remeshing and reassigning boundary conditions. Individual-body display is for inspection.

Separated parts and point-only or edge-only contacts remain independent. Each independent body needs sufficient constraints. If a saved study lacks its original solid snapshots, reimport the geometry before changing contacts. Save the study in the drawing and review conditions, mesh size and results together.

28 Mesh size and accuracy

A smaller mesh size represents detail more closely but increases elements, nodes, calculation time, and memory. Refine progressively to check whether the quantities of interest stabilize, rather than starting with an extremely fine mesh.

2D and 3D differences

Item	2D	3D
New mesh	Eight-node quadrilaterals	Ten-node tetrahedra
Geometry	One outer boundary with holes, or a planar face	Closed solids
Local refinement	Up to 32 rectangular regions	Not supported in this version
Curves and curved surfaces	Subdivide an approximated boundary	Approximate the surface with triangles

2D quadrilateral regularization

Regularize 2D Quadrilaterals is True by default. A rectangle without holes uses a grid aligned with its edges. For rectangles, the final quadrilateral edge length is approximately one-sixth of the entered mesh size; do not assume the input equals the actual element size. Check the element count and display.

Shapes with holes, curves, or notches are meshed while preserving those features. To compare regularization, set it to False and regenerate. Regularization does not necessarily improve accuracy for every load case.

How to compare

1. Keep material, supports, and loads fixed and analyze with mesh size h.

2. Regenerate with a size smaller than h and apply the same total conditions at the same locations.

3. Record and compare displacement at the same point, stress in the evaluation region, total reactions, and element count.

4. Investigate further if differences are large. If only maximum stress keeps increasing near corners, point loads, or fixed ends, review the idealized geometry and conditions as well.

Remeshing changes node and edge-segment numbers. Redefine conditions instead of reusing old numbers directly.

29 Enter loads correctly

Total, distributed, line, and point loads are different quantities. Entering 100 can produce different total forces depending on the type.

Condition	Units	Example input and total force
Total load	N／kgf	100 N assigned to two faces together gives 100 N overall
Distributed load	N/mm²	0.5 N/mm² over 200 mm² gives 100 N
Line load	N/mm	5 N/mm along a 20 mm edge gives 100 N
Point load	N	100 N applied to one node

Total load is apportioned by face area, edge length, or vertex count. For 2D distributed loads, area is selected edge length × thickness. 1 kgf = 9.80665 N.

Example: a total downward load of 100 N

If downward is −Y in 2D, set total-load X = 0 and Y = −100. If downward is −Z in 3D, set X = 0, Y = 0, Z = −100. Check the displayed coordinate axes before entering values.

Component loads on faces and solids act in fixed directions. They are not automatically interpreted as pressures following each point's normal on a curved surface. Arrow length does not quantitatively represent force magnitude.

Check restraints

Fixed sets every translational component to zero; Prescribed Displacement specifies every translational component. Distinguish this from leaving an arbitrary single continuum axis free. Beam, truss, and frame degree-of-freedom checkboxes follow their respective window rules.

Fixing only a straight edge may leave rigid-body rotation about that line. Unbonded components can also move independently. Before increasing the time limit, check that restraints represent the actual supports.

Load rows at the same location are added. Delete the old row when replacing a condition.

30 Tutorial: compare beam results with known values

This example applies a downward central load of 100 N to a simply supported beam 1,000 mm long with a rectangular section 10 mm wide and 20 mm high. Values verify operation and units; they are not product allowable limits.

Inputs	Value
Young's modulus E	200,000 N/mm²
Beam length L	1,000 mm
Section area A	200 mm²
Second moment of area I	6,666.6667 mm⁴
Extreme-fiber distance c	10 mm
Start support	Pin at position 0
End support	Roller (Y fixed) at position 1,000
Point load	Position 500; Fx = 0, Fy = −100 N

1. Open Analysis > 2D Analysis > Beam Calculation. Check the supports/loads table to avoid duplicating initial conditions.

2. Check material E and section properties and enter the table values. If importing a CAD section, choose the bending axis corresponding to the 20 mm height.

3. Run Calculation and inspect deflection, bending moment, normal stress, and support reactions.

Reference values from beam theory

Euler–Bernoulli theory gives central deflection magnitude $PL^3/(48EI)$ = 1.5625 mm, maximum bending moment PL/4 = 25,000 N·mm, and maximum bending stress Mc/I = 37.5 N/mm². Each support reaction is 50 N upward.

A downward displacement component is negative, while displacement magnitude is nonnegative. If results differ greatly, check I's bending axis, the magnitude of E, mm⁴ units, duplicate loads, and support positions. Do not equate local stress from a 3D solid directly with this beam-theory value.

31 3D beam local axes and torsion constant

3D beam loads, restraints, and results use the displayed local beam axes even when the CAD axis is oblique. X points from start to end; Y and Z are the section's principal axes.

1. Select a planar section or outer and hole boundaries in CAD and run Import Selected CAD Section.

2. Specify the axis with Beam Axis: Selected Line or Beam Axis: Two CAD Points.

3. Check the section's major/minor axes and Section rotation. Rotation also changes the local axes used for loads and restraints.

4. Add fixed ends or Custom Restraint, point loads, moments, and distributed loads over intervals.

5. If torsional loading is present, enter the appropriate section torsion constant J before calculating.

J is not calculated automatically

Importing a CAD section calculates area, centroid, and second moments of area, but sets torsion constant J to 0 (unset). J is required for an Mx load. For a general section, do not replace J with Iy + Iz.

Quantity	Units and direction
Fx, Fy, Fz	N; local XYZ
Mx, My, Mz	N·mm; positive by the right-hand rule about each axis
Distributed load over an interval	N/mm; start/end positions and direction
Iy, Iz, J	mm^4
Restraints	Check X, Y, Z, Rx, Ry, and Rz

Apply the necessary Rx restraint to prevent free rigid-body torsional rotation. 3D beams include axial force, bending in two directions, and torsion, but exclude shear deformation, restrained warping, and section changes along the beam. Normal stress does not include torsional shear stress.

32 Truss/frame connections and results

Trusses use pin-jointed members carrying only axial force; frames use rigid-jointed beam elements that also transmit bending. Even an identical-looking structure gives different results under different joint assumptions.

Draw before importing

1. Draw member centerlines as lines or straight polylines. For 2D, place them in one plane parallel to model XY.

2. Split lines at intended joints and make endpoints coincide. Lines crossing mid-span are not connected automatically.

3. Select the centerlines, import into Truss/Frame, and check node numbers and connections.

4. Set material and section values, enter nodal restraints/loads, and calculate. Checked degrees of freedom are fixed; unchecked ones are free.

Item	Truss	Frame
Main transmitted forces	Axial force	Axial force, shear, bending; torsion in 3D
Rotational degrees of freedom	None	Present
Main results	Nodal displacement/reactions; axial force/stress	Nodal displacement/reactions; member-end forces
Deformed display	Straight lines joining displaced nodes	Same; not the deflection curve within each member

Forces use N and moments N·mm. Nodal inputs use model coordinates, unlike the local coordinates of a standalone 3D beam. Member-force i/j denotes the start/end of the member; those forces use member-local coordinates.

Configuration and size limits

Material, section area, second moments of area, and torsion constant are common to every member. The limit is 2,400 degrees of freedom. The function does not automatically assign different designed sections to individual members.

Maximum nodal displacement is a value at a node. If mid-span deflection or local section stress is needed, also consider beam calculation or an appropriate solid analysis.

33 Interpret steady-state thermal boundary conditions

Steady-state thermal analysis finds a state whose temperature no longer changes with time. Choose transient analysis to investigate cooling from an initial temperature.

Condition	Input	Meaning
Prescribed temperature	X：℃	Maintain the selected location at that temperature
Heat flux	X：W/m²	Positive inward; negative outward
Convection	X: W/(m² K); Y: ambient temperature in °C	Heat exchange according to the surface-to-ambient temperature difference
Insulation	No numeric input	Zero heat transfer

Unset boundaries are insulated if Apply convection to unset boundaries is False. The surroundings do not automatically become 20°C. Each independent region needs a temperature reference, such as a prescribed temperature or effective convection condition.

Tutorial: impose a left-to-right temperature difference

Set a rectangle's left edge to 100°C and right edge to 20°C, insulate the remaining edges, and use constant thermal conductivity. Generate the mesh, select the edges, register both temperature conditions, and calculate. Check for a continuous temperature gradient between the two sides.

If only one node is held at 100°C and everything else is insulated, the entire body may reach 100°C at steady state. This is not a calculation of the cooling process.

Thermal bonding between materials

Perfectly bonded faces connect without thermal resistance. Active bonded interfaces are excluded from bulk convection. If bonding is disabled, those faces are treated as external and receive bulk convection when enabled. Assign individual insulation where convection is unwanted.

2D convection acts on boundary edge length × thickness. It excludes the broad front and back faces of a plate. If that effect matters, consider a 3D model.

34 Tutorial: observe cooling in transient thermal analysis

Begin with a simple box to check overall cooling. These values illustrate operation only. Adjust material properties and heat transfer coefficients to the actual component, fluid, and flow velocity.

1. Select Transient Thermal Analysis and import the geometry.

2. Check thermal conductivity, density, and specific heat, then generate the mesh.

3. Set initial temperature 100°C, end time 60 s, and time step 1 s.

4. Enable Apply convection to unset boundaries. For this example, use ambient temperature 20°C and heat transfer coefficient 10 W/(m² K). Do not add prescribed temperatures for this natural-cooling example.

5. Run the analysis and move the time slider from 0 to 60 s to inspect temperature changes.

Setting	Meaning	Effect of a change
Initial temperature	Temperature at the start	Starting point for cooling/heating
Prescribed temperature	Temperature maintained throughout analysis	The target has the prescribed value from the starting time
End time	Duration of the physical process calculated	How many seconds into the process to solve
Time step	Time interval used in calculation	Temporal accuracy and calculation effort
Result storage count	Approximate number of recorded times	Display/output sampling and memory use
Playback (s)	On-screen playback duration	Does not change the calculated physical times
Calculation time limit	Maximum time allowed for processing	Separate from physical end time

If the entire body is insulated and has no heating conditions, its initial temperature is retained. Compare with a smaller time step and examine the temperature history. Material values and conditions remain constant; time tables, radiation, phase change, and temperature-dependent properties are outside this version's scope.

35 Distinguish displacement, stress, and reactions

Three checks to make first

1. Does the deformation direction agree with the assumed loads and supports? Distinguish exaggerated display deformation from actual displacement.

2. Do total external forces and reactions balance? For example, an external force of −100 N should correspond to approximately +100 N of total reaction in that direction.

3. Where are high stresses concentrated? Inspect the region of interest, not only the immediate vicinity of fixed ends or point loads.

Result display	Meaning
Displacement	mm; displacement magnitude. Display scale does not change calculated values
Von Mises stress	N/mm²; element equivalent stress, not an allowable-stress decision by itself
Reaction force	Force exerted by restraints; distinguish whole-model and selected-range totals
Normal strain	mm/mm; positive extension, negative contraction
Engineering shear strain	Shear in the specified plane; distinct from normal strain
Equivalent elastic strain	Currently defined as von Mises stress / E; not plastic strain

Colors and maximum values

Smooth display interpolates between nodes and averages quantities such as stress and strain for visualization. A smooth appearance does not improve mesh accuracy. The legend uses displayed values and may differ from results including internal elements. Use numeric export to inspect all required element values.

Original Outline compares the undeformed shape in displacement display. Do not infer displacement from stress colors alone. Clear Selection or Esc removes distracting yellow highlights without deleting conditions or results.

What to record for comparison

Record material, geometry revision, mesh size and element count, restraints, total load and direction, displayed component, units, deformation scale, and time for thermal results. The same red color can represent different values if legend ranges differ between figures.

36 Save and export conditions and results

Saving analysis conditions and saving the drawing file are two separate steps. Run Save/Store Conditions in Drawing, then save the JDC normally.

Save or export	Details	Reuse
Save/Store Conditions in Drawing	Material values, mesh/members, boundary conditions, etc.	Open JDC and recalculate in the analysis window
Export Results: JSON	Conditions and numeric results	Records and external data processing
Nodal Reactions CSV	Node coordinates, reaction components, etc.	Check totals and selected regions
Element Strain CSV	Components and equivalent elastic strain	Inspect element results, including the interior
Transient Temperature CSV	Nodal temperatures at saved times	Check temperature histories
Save Image / Print	Current viewpoint, displayed quantity, legend, etc.	Evaluation documents
Place Beam Results in CAD	Selected result diagram, scale, maximum values, etc.	Explanatory diagrams in the drawing

One condition set is stored per 2D/3D analysis category. For case comparisons, use separate JDC and result files with descriptive case names. Steady-state/transient thermal conditions are shared within 2D and within 3D, so use separate files to retain independent cases.

Saved materials

Analysis conditions retain the material values present when saved. Later changes to the shared material list do not automatically overwrite saved conditions. To use updated values, reselect the material and recalculate.

Prepare the display before exporting

Fit the model with Zoom Extents and choose the result and scale. Image output excludes yellow selection highlights and operating prompts but includes the legend. For transient results, select the required time. Numeric export is separate from saving playback as a video; video export is not supported in this version.

37 Analysis troubleshooting checklist

Symptom	Main checks	Action
Cannot select a face for a condition	Check mesh generation and selection mode	Generate the mesh and select Face/Edge/Vertex
2D import fails	Closed outline, coplanarity, multiple outer boundaries	Use one outer boundary with holes
3D mesh cannot be generated	Open geometry, overlapping volumes, tiny features	Inspect the source BRep and component positions
Calculation fails from insufficient restraints	Rigid-body translation/rotation; independent components	Define actual supports and bonding
Load exceeds expectations	Duplicate rows; confusion between distributed and total loads	Compare total loads with reactions
Displacement differs greatly	E, section I, thickness, units, axes	Check mm/N units and bending axis
Maximum stress rises as the mesh is refined	Corners, point loads, fixed ends	Review the evaluation region and actual contact area
Uniform temperature throughout	Insulation, prescribed temperatures, steady-state mode	Use transient analysis for the cooling process
No cooling in transient analysis	Disabled convection, ambient temperature, fixed temperature	Check bulk/individual convection and initial temperature
CAD edits do not change results	Previously imported geometry	Reimport, remesh, and redefine conditions
Old values appear after saving	Check whether JDC was saved after storing conditions	Store conditions and save the drawing
Calculation takes too long	Element count, time limit, model size	Remove irrelevant details and analyze progressively

Increasing only the time limit or mesh density does not fix an invalid model. Check geometry, materials/units, bonding, restraints, and loads in that order to narrow down the cause.

Part 5 IFC Components and Revit Families

This part explains geometry, classification, properties, and connection ports, followed by conversion from IFC to RFA.

01 Workflow from IFC to Revit family 349

Starting IFC library export

Choose Insert > Export > Export IFC Library. This command is available in STD and Pro. Select the solids that make up one component, then configure its classification, properties, and connection ports.

Part 5 covers curved BRep in IFC4, multiple solids and blocks, and RFA creation using Dynamo for Revit 2026/2027. Unlike ordinary CAD export, this command requires component classification and, where applicable, equipment connection ports to be configured before export.

02 Choose geometry, base point, and output format 350
03 Classification, standard properties, and additional specifications 351
04 Define connection ports from 3D geometry 352
05 Cold water, hot water, sanitary, and gas ports 353
06 Configure a power IN connector 354
07 Save settings and export IFC 355
08 Create RFA with Dynamo Player 356
09 Check properties and connectors in Revit 357
10 Conversion scope and troubleshooting 358
Classification list: 58 categories and their IFC4 mappings 359

01 Workflow from IFC to Revit family

Export IFC Library assigns classification, product information, and connection ports to one component created in JDesignCAD. IFC export requires STD or Pro. Conversion to RFA uses Revit 2026 or 2027 and its bundled Dynamo.

Stage	Function used	Create or verify
1	JDesignCAD 3D drawing	Component solids, colors, and base point
2	Export IFC library	Classification, properties, and port settings
3	IFC2x3/IFC4 export	An IFC file for one component
4	Dynamo Player in Revit	Select IFC → save RFA
5	Revit	Family types, materials, connectors, and project connections

One component with multiple solids

Separate solids for a tabletop, legs, and modesty panel can be exported as one component. They do not need to be Boolean-unioned into a single solid. One IFC file contains one component occurrence and type, with multiple solids representing its geometry.

IFC Library is the name of this component-export function. It does not cover every IFC class, create an IfcProjectLibrary managing multiple product numbers, or generate Revit parametric modeling history. The supplied catalog contains 58 categories.

Family geometry and properties

The RFA created by Dynamo is a loadable family with fixed geometry. It retains manufacturer, model, and specifications, and converts supported ports to native Revit connectors. Changing nominal width or height values does not resize the body.

Use it for static components such as furniture, equipment placement, and supported piping/power connections. Specialized functions such as wall openings, light sources, and structural analytical models must be created separately in Revit.

02 Choose geometry, base point, and output format

1. Select every solid belonging to the component. Supported blocks containing solids can also be selected.

2. Choose Insert > Export > Export IFC Library.

3. In Basic Information / Base Point, set the component name, model, manufacturer, and placement base point.

4. Choose the output format. To transfer curved geometry as BRep, choose IFC4 Curved BRep.

Format	Geometry representation	When to choose
IFC4 Curved BRep	IfcAdvancedBrep	Components with curved surfaces and edges; the standard choice
IFC4 Faceted BRep	IfcFacetedBrep	When the receiving application requires faceted geometry
IFC2x3 Faceted BRep	IfcFacetedBrep	When IFC2x3 is required for delivery

For curved BRep, surfaces or boundaries that cannot be represented directly may be converted to bounded NURBS passing a 0.001 mm accuracy check. This does not mean that every curved surface is converted into a mathematically identical representation.

If conversion fails, the exporter does not silently switch to faceting or omit faces. When choosing faceted output, set the subdivision accuracy and check closure, face orientation, and volume before exporting.

Exporting dynamic blocks

Set the required dimensions, lookup selection, and visibility state before selecting the block. The current visible geometry is evaluated as a component containing multiple solids. Exploding beforehand is normally unnecessary. Dynamic action definitions and hidden alternative configurations are not transferred to Revit.

Choose a base point convenient for placement in the receiving application. Dynamo preserves exported IFC coordinates and does not automatically recenter the geometry.

03 Classification, standard properties, and additional specifications

Selecting a category displays the correspondence between property labels and IFC attribute names. Follow the category guidance and enter actual product data. Not every displayed field is mandatory under the IFC standard; also check the recipient's requirements.

Tab	What to enter	Example
Basic Information / Base Point	Component name, model, manufacturer, description, etc.	OA Table, model 001
Category Standard Properties	Specifications associated with the category	Nominal height, nominal length, main color
Additional Product Specifications	Names and values/units absent from the standard fields	Tabletop material: melamine-faced board
Connection Ports IN / OUT	Position, direction, domain, system, etc.	Cold Water IN, Power IN

Example furniture dimensions and color

For Furniture / Table, enter nominal height 720 mm, nominal length 1,800 mm, and nominal depth 800 mm. These are product specifications, not dimensions automatically measured from the outline. Enter values matching the actual model.

Enter text such as White in Main Color. This text does not change the solid's display color. Check geometry colors and transparency in the CAD face/material settings.

Examples of additional specifications

Specification names can be entered in Japanese. Use clear units and descriptions, such as Rated Power Consumption = 100 W or Mounting Method = Floor-standing. The three Dynamo power-specification fields require exact names and numeric-only values; follow the power-connector section below.

Properties whose standard definitions differ in IFC2x3 are supplemented with unit-bearing custom properties beginning with JDC_, with a notification on completion. The receiving application may display different names or locations.

04 Define connection ports from 3D geometry

Acquire position, normal, and opening dimensions from geometry, then complete the necessary numeric values. Product meaning such as IN/OUT, system, and nominal diameter cannot be determined from geometry alone and must be checked.

1. Under Connection Ports IN / OUT, click Add from 3D Geometry. To revise an existing row, use Set Selected Port from 3D Geometry.

2. Click the connection face. It defines the direction outward from the component.

3. Click the opening outline. A circle provides center and diameter; a rectangle provides center, width, height, and section rotation.

4. Back in the table, check the port name, domain, system, flow, nominal diameter, and dimensions. Correct numeric values as needed.

5. If the axis is reversed, use Reverse Selected Port Direction. Also check the position relative to the geometry and base point.

Selected geometry	Acquired data and notes
Circular/arc opening	Center and inner/outer diameter
End face bounded only by two concentric circles	Both inner and outer diameters
Flange face with several holes	Dimensions of the selected outline; select the intended opening
Rectangle made of straight lines	Center, width, height, and section rotation
Other outlines	Combine Point on Face with numeric input

Positions are WCS coordinates in mm; directions are vectors pointing outward from the component. This is separate from flow IN/OUT. Selecting IN does not require the user to reverse the outward axis inward.

Esc cancels while retaining existing settings. All acquired values remain numerically editable. Revit connector conversion requires the port to lie on a supported planar end face or at the center of a circular opening.

05 Cold water, hot water, sanitary, and gas ports

IFC port definitions and the scope of Dynamo conversion to Revit connectors differ. For piping connectors with the supplied Dynamo, use the following settings.

Item	Setting
Domain	PIPE
Cross-section	CIRCULAR
Nominal diameter	Enter in mm
Cold-water system	DOMESTICCOLDWATER
Hot-water system	DOMESTICHOTWATER
Sanitary system	SANITARY
Gas system	GAS; maps to Other piping in Revit
Flow	IN = SINK; OUT = SOURCE; bidirectional = SOURCEANDSINK

Example component with two ports

Create a simple valve and place Cold Water IN and Cold Water OUT on its two planar ends. For example, use domain PIPE, the same cold-water system, circular sections, and nominal diameter 25 mm; distinguish only the IN/OUT flow. Each axis points outward from its own end face.

Dynamo creates an internal link for pumps and valves with two ports of the same system. It does not infer internal routes with three or more ports or connections between different systems. Replace NOTDEFINED flow with an explicit setting.

Check in Revit

Open the RFA and select connector symbols to check system, flow, and position. Family-type connection diameters are associated with connectors, but the body's hole diameters remain fixed. Load the family into a project and test actual connections to compatible pipes.

For gas, prepare an appropriate project system under Other piping. Release validation samples cover cold water, gas, and power. Hot water and sanitary ports are also conversion targets, but verify their connections under your intended conditions.

06 Configure a power IN connector

The supplied Dynamo graphs creates electrical power IN ports as Revit Power - Balanced connectors. Power outputs, control/communications, duct, and conduit connectors are outside its scope.

Port field	Setting
Port name	A distinctive name, such as Power IN
Domain	CABLE
System	ELECTRICAL
Flow	SINK（IN）
Position and axis	On a planar end face, pointing outward from the component

Three fields required in Additional Product Specifications

Specification name (enter exactly as shown)	Example value	Input rule
電源電圧 (V) [Supply Voltage]	100	Number only; do not append V
電源極数 [Number of Poles]	1	Number only
電源皮相電力 (VA) [Apparent Power]	100	Number only; distinguish VA from W

These values illustrate operation; use the actual product ratings. Enter the Japanese field names above exactly, without the bracketed English explanations. For multiple power ports, prefix each of the three names with the port name, for example Power IN / 電源電圧 (V), and define all three values for each port.

After conversion to Revit

Select the connector symbol and check voltage, pole count, apparent power, and system. Parameters prefixed with the port name are also added to the family type and associated with the connector. Create a power circuit after loading into a project.

Panel distribution systems, panel schedules, load classifications, and destination panels are not configured automatically. Creating a connector does not complete the project's electrical setup. Configure it for the receiving project.

07 Save settings and export IFC

1. Check classification, basic information, properties, geometry base point, and port positions/directions.

2. Select the review confirmation checkbox, click Export IFC, and choose the destination.

3. Review completion messages. Inform the recipient of details such as supplemented IFC2x3 properties.

4. Use Save Settings in Drawing and save the JDC normally to retain the source for future use.

Where settings are stored

Component settings are stored on the first selected element. For a multi-solid component, select the same complete set next time. Saving settings in the drawing supports Undo/Redo. Exporting an IFC file and saving the source JDC are separate operations.

If export fails

Details	What to check
No solid recognized	Check for closed solids rather than only lines or open faces
Geometry validation fails	Check face boundaries, orientation, volume, and previous Boolean results
Curved BRep conversion fails	Inspect the geometry; choose another format only if it meets recipient requirements
Property input error	Check numbers/units, selection fields, and required settings
Connection position differs from expectations	Check WCS, base point, outward axis, and units

After editing geometry, recheck saved ports. Do not assume semantic connection information always follows moved holes or end faces automatically. Reacquire the geometry or adjust numeric values.

Connections between IFC components are exported as unconnected. Defining ports within a component is separate from deciding which pipes or equipment they connect to in the building.

08 Create RFA with Dynamo Player

The installed Revit/Dynamo folder, or the standalone ZIP folder, contains the converter files and 使い方.txt (usage instructions). A separate Dynamo graph is not needed for each component.

Revit version	Graph to run
Revit 2026	JDesignCAD_IFC からファミリ_Revit2026.dyn
Revit 2027	JDesignCAD_IFC からファミリ_Revit2027.dyn

Initial setup

1. If needed, copy the distribution folder to your working folder.

2. Open a project in Revit; an empty project is sufficient.

3. Open Manage > Dynamo Player and add/select the folder containing the .dyn files.

4. If a trust prompt appears, verify the source before allowing execution.

Each conversion

1. Run the converter matching the year version of the current Revit application.

2. In Select a JDesignCAD IFC Component, choose the exported IFC file.

3. In Save Family, specify the RFA name and destination.

4. On completion, open the saved RFA or load it into a project.

Select an .rft only if a family template cannot be found automatically. Use a non-hosted Metric Generic Model or Furniture template. Do not use templates with wall/ceiling hosts or existing geometry.

No additional product add-in or Dynamo package is required. Revit LT and standalone Dynamo are not supported. You do not need to link IFC, bind it, or ungroup it first.

The 2026 graph uses CPython3; the 2027 graph uses PythonNet3. Execution stops in a mismatched year version. RFA saved by 2027 cannot be opened in 2026. To distribute to both, create in 2026 and upgrade for use in 2027.

09 Check properties and connectors in Revit

Inspect in the Family Editor

Open the RFA and Family Types to inspect component name, IFC classification, manufacturer, model, standard properties, and additional specifications. Manufacturer and model are also assigned to Revit's standard Manufacturer and Model fields.

After loading into a project, select a placed component and open Edit Type. Check placement-specific information in the Properties palette and shared specifications in the type settings.

Item	Transferred data and notes
IFC property sets	Retain parameters with the original property names
Standard properties	Japanese-labeled properties are also added
Length	Revit Length data type
Boolean values	Yes/No data type
Other values	Text including units
Face colors and transparency	Shading materials read by Revit
Textures	Not created automatically

Select a connector

Select a connector symbol in the RFA and check its system, flow, and description in Properties. Family Types parameters such as Cold Water IN Connection Diameter and Power IN Voltage are associated with the connectors.

Changing connection diameter does not change the fixed body's hole diameter or outer shape. Original IFC specification properties record values at import and do not automatically follow connector edits. Before delivery, check that body geometry, connectors, and specifications agree.

When inspecting an IFC link

Type properties and project information for the entire link are different from the component's family type. To use a loadable family, create an RFA with the Dynamo workflow in this part and inspect that RFA.

10 Conversion scope and troubleshooting

Furniture, plumbing fixtures, mechanical equipment, and other supported categories are assigned to corresponding Revit categories. Categories for which the converter has no specialized support, such as walls, floors, roofs, straight pipes/ducts, and beams, are saved as fixed Generic Model families with a notice. IFC classification remains in the properties.

Target	Scope of The supplied Dynamo graphs
Component geometry	Geometry Revit can read as solids; multiple solids supported
Component specifications	Basic information, standard properties, and additional specifications
Piping ports	Circular cold-water, hot-water, sanitary, and gas ports
Power ports	Power - Balanced for CABLE / ELECTRICAL / SINK
Other ports	Stop and notify if an unsupported port is present
Doors and windows	No wall-cutting host behavior is created
Light sources and structural analytical models	Not created
Target IFC	One component exported by JDesignCAD; not a general building converter

Checks before and after saving

The converter checks solid count, volume, position, properties, and port count, position, direction, and system. It stops for missing solids, meshed geometry, unresolved color mapping, property-transfer failures, and similar issues. It does not save an apparently complete RFA with silently omitted content.

The source IFC is copied to a temporary folder for processing. No links or geometry are added to the open project. A validated temporary RFA is moved to the destination, so ordinary conversion failures or cancellation leave an existing RFA unchanged. Saving an RFA file is not covered by the project's Undo operation.

If a problem occurs

Check the error Details, Revit year version, .dyn used, target IFC, and intended port name. For connectors, first verify a planar end-face location and explicit axis, system, and flow. If removing unsupported ports to create geometry only, treat the result as a component without those connection functions.

Part 5 Classification Reference

The following lists identify IFC4 classifications. IFC2x3 export uses corresponding occurrences and types, whose names appear in the export dialog.

Classification List 01: Categories and IFC4 Mapping..... 360
Classification List 02: Categories and IFC4 Mapping..... 361
Classification List 03: Categories and IFC4 Mapping..... 362
Classification List 04: Categories and IFC4 Mapping..... 363
Classification List 05: Categories and IFC4 Mapping..... 364
Classification List 06: Categories and IFC4 Mapping..... 365

Classification List 01: Categories and IFC4 Mapping

These are the 58 categories included in the export dialog, with their IFC4 entity and predefined type. Check the selected category's standard properties in the dialog and supplement additional specifications and connection information.

Category	IFC4 entity / type	Example specification and port checks
Sanitary / Toilet	IfcSanitaryTerminal / TOILETPAN	Cold-water IN, sanitary OUT; power for a washing seat where applicable
Sanitary / Urinal	IfcSanitaryTerminal / URINAL	Cold-water IN, sanitary OUT; automatic-flush power where applicable
Sanitary / Washbasin	IfcSanitaryTerminal / WASHHANDBASIN	Drain OUT; cold/hot-water IN if the component includes a tap
Sanitary / Sink	IfcSanitaryTerminal / SINK	Drain OUT; cold/hot-water IN if a tap is included
Sanitary / Bathtub	IfcSanitaryTerminal / BATH	Hot-water IN, drain OUT; add reheating circulation as required by the product
Sanitary / Shower	IfcSanitaryTerminal / SHOWER	Cold/hot-water IN; mixer and outlet depend on the actual assembly
Sanitary / Cistern	IfcSanitaryTerminal / CISTERN	Cold-water IN, flushing-water OUT to the toilet
Sanitary / Toilet Seat	IfcSanitaryTerminal / WCSEAT	None for a standard seat; water IN and power for washing seats as applicable
Piping Equipment / Circulating Pump	IfcPump / CIRCULATOR	Suction IN, discharge OUT, power; drain if required
Piping Equipment / End-Suction Pump	IfcPump / ENDSUCTION	Suction IN, discharge OUT, power

This table lists IFC export classifications, not Revit connector support. Even where duct or control/communications ports can be defined in IFC, bundled Dynamo conversion to native Revit connectors is limited to sections 05, 06, and 10 of this part.

Classification List 02: Categories and IFC4 Mapping

These are the 58 categories included in the export dialog, with their IFC4 entity and predefined type. Check the selected category's standard properties in the dialog and supplement additional specifications and connection information.

Category	IFC4 entity / type	Example specification and port checks
Piping Equipment / Submersible Pump	IfcPump / SUBMERSIBLEPUMP	Discharge OUT and power; distinguish an open suction inlet from a pipe connector
Piping Equipment / Straight Pipe	IfcPipeSegment / RIGIDSEGMENT	Normally two ports; bidirectional/undefined flow can be selected for a pipe without a fixed direction
Piping Equipment / Flexible Pipe	IfcPipeSegment / FLEXIBLESEGMENT	Normally two ports; retain flexible-body position separately from end directions
Piping Equipment / Elbow	IfcPipeFitting / BEND	Two ports; set position, axis, and connection method for each
Piping Equipment / Tee or Branch Fitting	IfcPipeFitting / JUNCTION	Normally three or more ports; set flow from the actual system
Piping Equipment / Reducer	IfcPipeFitting / TRANSITION	Two ports; retain different diameters and eccentric offset
Piping Equipment / Flange Fitting	IfcPipeFitting / CONNECTOR	Define piping connection faces; do not treat bolt holes as ports
Piping Equipment / Shutoff Valve	IfcValve / ISOLATING	Normally two ports; specify ball, gate, or other construction as properties
Piping Equipment / Check Valve	IfcValve / CHECK	Inlet IN, outlet OUT; confirm the intended one-way use
Piping Equipment / Pressure-Reducing Valve	IfcValve / PRESSUREREDUCING	High-pressure IN, low-pressure OUT

This table lists IFC export classifications, not Revit connector support. Even where duct or control/communications ports can be defined in IFC, bundled Dynamo conversion to native Revit connectors is limited to sections 05, 06, and 10 of this part.

Classification List 03: Categories and IFC4 Mapping

These are the 58 categories included in the export dialog, with their IFC4 entity and predefined type. Check the selected category's standard properties in the dialog and supplement additional specifications and connection information.

Category	IFC4 entity / type	Example specification and port checks
Piping Equipment / Tap	IfcValve / FAUCET	Cold/hot-water IN, discharge OUT; distinguish pipe ports from free discharge
Piping Equipment / Water Storage Tank	IfcTank / STORAGE	Inlet IN, outlet OUT; drain, overflow, and vent as required
Piping Equipment / Water Boiler	IfcBoiler / WATER	Water inlet/outlet; add fuel, exhaust, and power according to the model
HVAC / Air Handling Unit (AHU)	IfcUnitaryEquipment / AIRHANDLER	Supply, return, outside, and exhaust air; water and drain connections where applicable
HVAC / Packaged Air Conditioner	IfcUnitaryEquipment / AIRCONDITIONINGUNIT	Refrigerant piping, drain, and power; duct connections depend on the model
HVAC / Axial Fan	IfcFan / PROPELLORAXIAL	Suction IN, discharge OUT, power; distinguish open ends
HVAC / Straight Duct	IfcDuctSegment / RIGIDSEGMENT	Normally two ports; retain width, height, and section rotation for rectangles
HVAC / Duct Bend	IfcDuctFitting / BEND	Normally two ports; set center, direction, and section for each
HVAC / Damper	IfcDamper / CONTROLDAMPER	Normally two ports; actuator power/control where applicable
HVAC / Diffuser	IfcAirTerminal / DIFFUSER	Duct connection; verify supply/extract use from the system

This table lists IFC export classifications, not Revit connector support. Even where duct or control/communications ports can be defined in IFC, bundled Dynamo conversion to native Revit connectors is limited to sections 05, 06, and 10 of this part.

Classification List 04: Categories and IFC4 Mapping

These are the 58 categories included in the export dialog, with their IFC4 entity and predefined type. Check the selected category's standard properties in the dialog and supplement additional specifications and connection information.

Category	IFC4 entity / type	Example specification and port checks
HVAC / Grille	IfcAirTerminal / GRILLE	Define the connection; do not infer supply/return from appearance alone
Fire Protection / Sprinkler Head	IfcFireSuppressionTerminal / SPRINKLER	Pipe IN; do not invent a pipe port at the spray outlet
Electrical / Distribution Board or Panelboard	IfcElectricDistributionBoard / DISTRIBUTIONBOARD	Define supply, outputs, and control as needed; distinguish each port's role
Electrical / Power Outlet	IfcOutlet / POWEROUTLET	Electrical connection; distinguish circuit connectivity from plug sockets
Electrical / Light Fixture (Point Source)	IfcLightFixture / POINTSOURCE	Power; dimming/control lines where applicable
Electrical / Cable Tray	IfcCableCarrierSegment / CABLETRAYSEGMENT	Mechanical tray connections; do not assign power IN/OUT indiscriminately
Electrical / Conduit	IfcCableCarrierSegment / CONDUITSEGMENT	Normally two ports; distinguish conduit connectivity from electrical circuits
Furniture / Chair	IfcFurniture / CHAIR	Normally none; add only for products with powered mechanisms or similar features
Furniture / Table	IfcFurniture / TABLE	Normally none; built-in power where applicable
Furniture / Desk	IfcFurniture / DESK	Normally none; configure built-in outlets separately

This table lists IFC export classifications, not Revit connector support. Even where duct or control/communications ports can be defined in IFC, bundled Dynamo conversion to native Revit connectors is limited to sections 05, 06, and 10 of this part.

Classification List 05: Categories and IFC4 Mapping

These are the 58 categories included in the export dialog, with their IFC4 entity and predefined type. Check the selected category's standard properties in the dialog and supplement additional specifications and connection information.

Category	IFC4 entity / type	Example specification and port checks
Furniture / Bed	IfcFurniture / BED	Normally none; power for electric/medical beds depends on the product
Furniture / Filing Cabinet	IfcFurniture / FILECABINET	Normally none
Furniture / Shelf	IfcFurniture / SHELF	Normally none
Furniture / Sofa	IfcFurniture / SOFA	Normally none; powered mechanisms where applicable
Furniture / Other Furniture	IfcFurniture / USERDEFINED	Depends on use; always record a descriptive name for USERDEFINED
Structural / Beam	IfcBeam / BEAM	No MEP ports; structural joints and analytical endpoints are separate
Structural / Column	IfcColumn / COLUMN	No MEP ports; check LoadBearing for structural-column use
Structural / Brace	IfcMember / BRACE	No MEP ports; end joints are structural relationships
Structural / Connection Plate	IfcPlate / SHEET	No MEP ports; do not interpret holes as piping ports
Structural / Bolt	IfcMechanicalFastener / BOLT	No MEP ports; mechanical fastening information

This table lists IFC export classifications, not Revit connector support. Even where duct or control/communications ports can be defined in IFC, bundled Dynamo conversion to native Revit connectors is limited to sections 05, 06, and 10 of this part.

Classification List 06: Categories and IFC4 Mapping

These are the 58 categories included in the export dialog, with their IFC4 entity and predefined type. Check the selected category's standard properties in the dialog and supplement additional specifications and connection information.

Category	IFC4 entity / type	Example specification and port checks
Structural / Isolated Footing	IfcFooting / PAD_FOOTING	No MEP ports; relationships to supported members are separate
Architectural / Wall	IfcWall / STANDARD	Normally no MEP ports; window/door openings and joints are separate relationships
Architectural / Floor Slab	IfcSlab / FLOOR	Normally none; openings and supporting-member relationships are separate
Architectural / Roof Slab	IfcSlab / ROOF	Normally none; distinguish the whole roof from its constituent slabs
Architectural / Door	IfcDoor / DOOR	Normally no MEP ports; power/control for automatic doors where applicable
Architectural / Window	IfcWindow / WINDOW	Normally none; power/control for powered opening where applicable
Architectural / Railing	IfcRailing / HANDRAIL	Normally none; retain posts and mounting components
Other / Unclassified Building Component	IfcBuildingElementProxy / USERDEFINED	Only if needed; first look for an appropriate dedicated IFC category

This table lists IFC export classifications, not Revit connector support. Even where duct or control/communications ports can be defined in IFC, bundled Dynamo conversion to native Revit connectors is limited to sections 05, 06, and 10 of this part.

Part 6 Installation and Updates

Use the current installer or Check for updates to update the application. Save drawings and component masters before updating.

Update with the installer

1. Save open drawings and close JDesignCAD.

2. Run the supplied JDesignCAD-1.1.4-Setup.exe.

3. Follow the installer to completion, then start the application and check that the title reads JDesignCAD Ver1.1.4.

Using the ZIP distribution

The ZIP contains the installer set. Extract everything and run Setup.exe from the extracted folder, not from inside the ZIP. You do not need to install both EXE and MSI separately. This is not an installation-free portable edition.

When automatic updates are unavailable

Use Check for Updates on the File screen to view update information. If automatic updating is unavailable, update using the newly distributed installer.

Save editable source data

Keep dynamic-block settings and analysis-condition masters in JDC. Save DWG, DXF, or other exchange files under separate names and verify that the required geometry and behavior are retained.

This manual describes JDesignCAD 1.1.4. Screenshots and analysis examples illustrate operation. See the relevant chapters for analysis assumptions and exchange-format limits.

01 Workflow reference guide

Domain	Current capabilities	Manual reference
Display language	Japanese/English, startup panel, and main dialogs	Part 1, section 25
Model units	mm, cm, m, km, inch, feet; preserve physical size	Part 1, sections 26–28
Data exchange	SXF, import units, STEP improvements	Part 2, SXF input/output; Part 1, section 28
Clothoids	Standard curves, basic IP form, cross-section lines, etc.	Part 2, Clothoids
Modeling	Multiple-face editing, hole re-editing, slicing improvements, etc.	Part 2, Face editing, holes and slicing
Components and layers	Layer 0 inheritance and UCS operations	Part 1, Layers and coordinate systems
Dynamic blocks	3D face actions, multiple solids, formulas, and grips	Part 3, sections 24–33
Analysis	Individual materials, perfect bonding, and detailed condition/result guidance	Part 4, sections 25–37
IFC and Revit	IFC4 curved BRep, properties/ports, Dynamo 2026/2027	Part 5
File preview	Reflects face transparency settings	Startup panel and preview

Scope of this manual

Basic operations and feature instructions are integrated here. Separate supplements for earlier releases are not required.

Analysis conditions, illustrated values, and component dimensions are learning examples. For actual use, configure them for the product specifications and real mounting, loads, materials, and heat-transfer conditions. Unsupported specialized functions and exchange-format limits are stated in their respective chapters.

02 Managing distribution files and source data

Distribution files

File or location	Use
JDesignCAD-1.1.4-Setup.exe	Install/update JDesignCAD
ZIP of the installer set	Extract all files and run Setup.exe
Installed Revit/Dynamo folder	2026/2027 .dyn files and usage instructions
JDesignCAD-IFC-RevitDynamo-1.1.4.zip	Standalone distribution of the Revit converter
JDC	Editable masters for drawings, components, and analysis conditions
IFC	Component exchange data with classification, properties, and ports
RFA	Fixed-geometry family for the Revit year version used

Retaining working data

Use JDC as the master for dynamic blocks and 3D face actions. Exchange formats may not preserve all actions or history. To revise IFC/RFA later, retain the JDC containing original geometry and port settings for easier updates.

Match each analysis case's JDC with its result exports and record geometry revision, material, restraints/loads, mesh, and time. Images alone do not retain enough information to recalculate.

Checks after updating

Restart JDesignCAD and verify the version. Restart after changing display language as well. Choose the Dynamo graph matching the Revit year version and organize folders to avoid using an older file accidentally.

Updating the application does not recover geometry lost in an older version or editing information already reduced to lines/text in exchange files. Keep pre-update files and backups.

Part 7 Sheet metal command reference

Sheet metal requires STD/Pro. Set thickness/bending rules and work on a base plate or recognized solid. Heavy calculations run on Apply; successful results are applied as one Undo step.

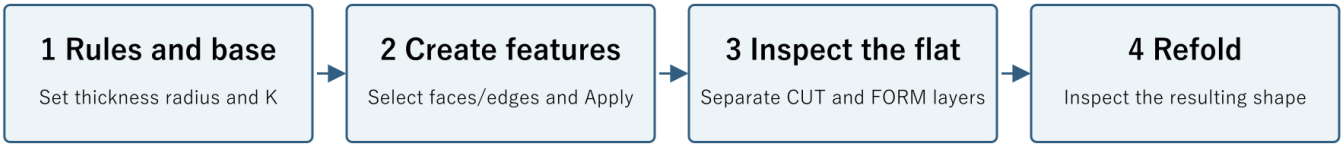


Figure 8 Typical sheet-metal workflow.

Unfolding formed features restores the reference flat and creates flat-pattern representations. It does not simulate material flow, thinning, failure or manufacturing blank size. Keep louver/bridge forming outlines separate from cutting lines.

Sheet metal dialogs

Sheet-metal commands calculate when you choose Apply. A successful change is one Undo step. Save in JDC to retain thickness, bend and flat-pattern information.

Base plate

Select one closed profile, set thickness t and its direction, then click Apply.
On the XY work plane, Positive goes up (+Z); Negative goes down (-Z).
Symmetric adds t/2 per side, total t. The original profile remains.

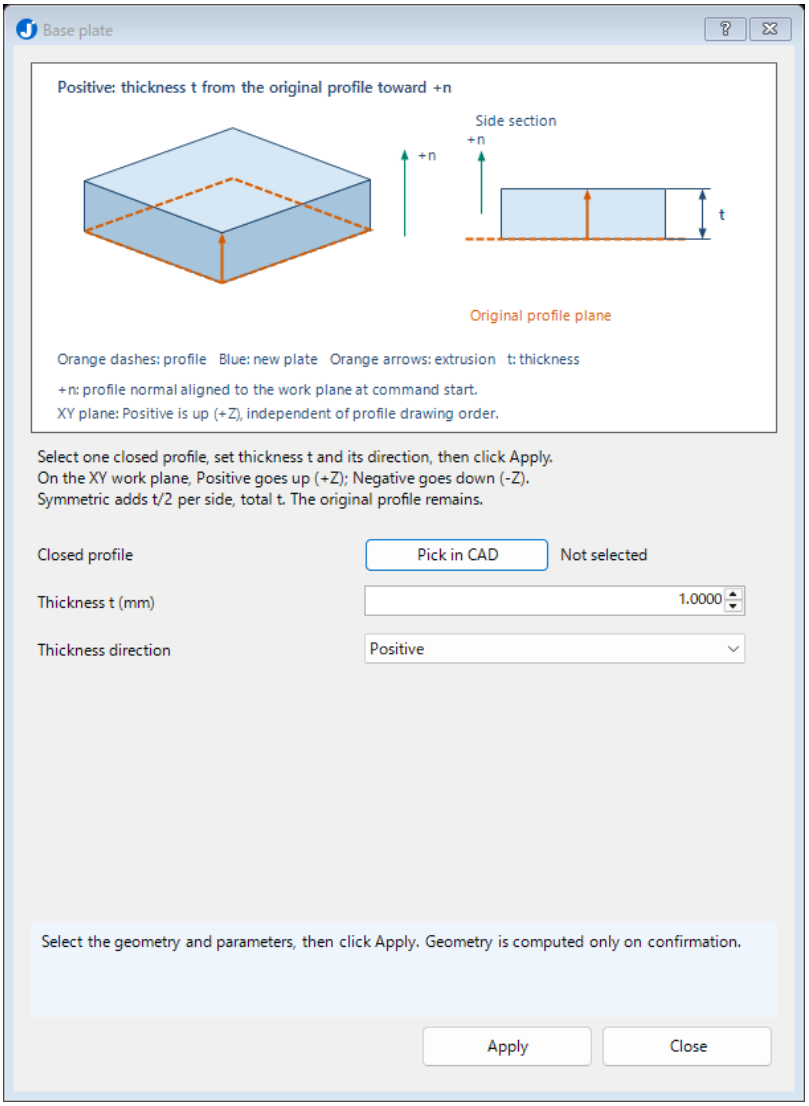


Figure 9 Current dialog. Set the target and inputs, then choose Apply.

Bend along line

Select the solid, fixed face, then bend line. Looking at that face, the right side of start→end bends.
Existing flanges on the moving side travel with it. Holes in the bend strip, existing curved bands and colliding bends are not supported.

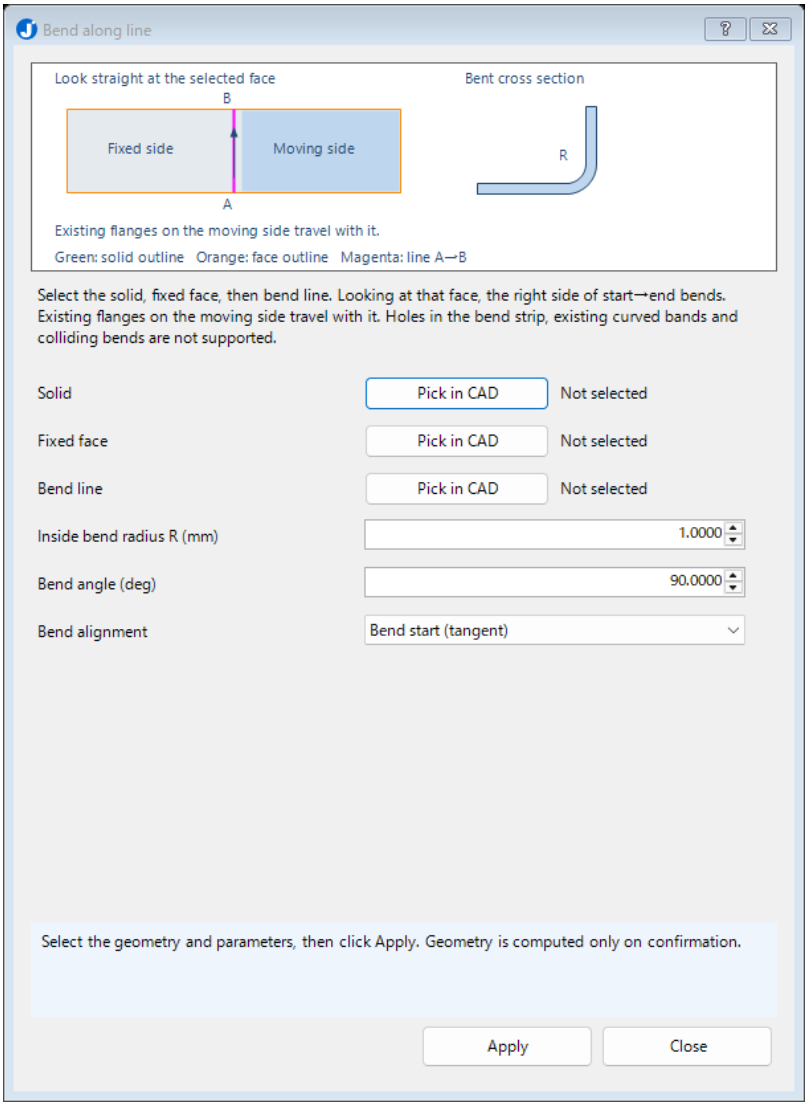


Figure 10 Current dialog. Set the target and inputs, then choose Apply.

Bend manager

Select a B number to suppress, resume, delete or change its parent. Delete also removes dependent flanges and forms. A parent change must retain connected material. Geometry is checked on Apply.

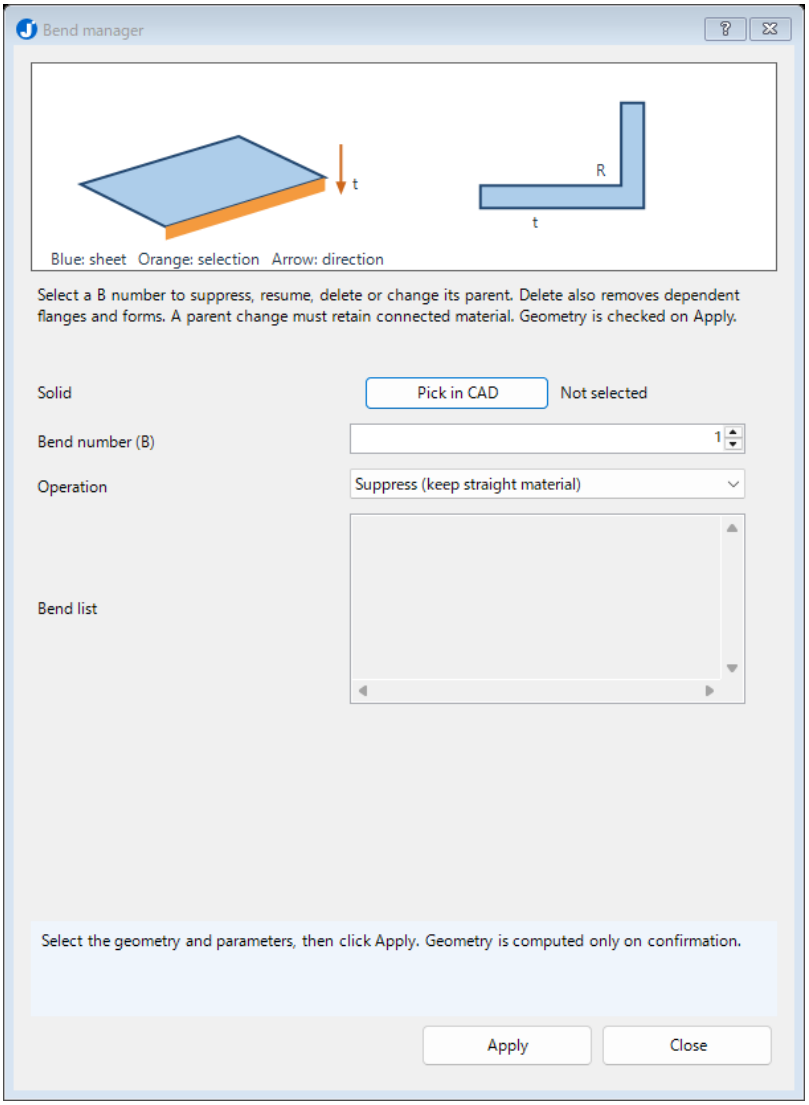


Figure 11 Current dialog. Set the target and inputs, then choose Apply.

Bend relief

Select a bend-end arc, set width w, depth d and shape, then Apply.
Create a relief or replace a registered one. None removes relief at that end.
Existing relief dimensions are not loaded automatically. Enter the desired values. See [?].

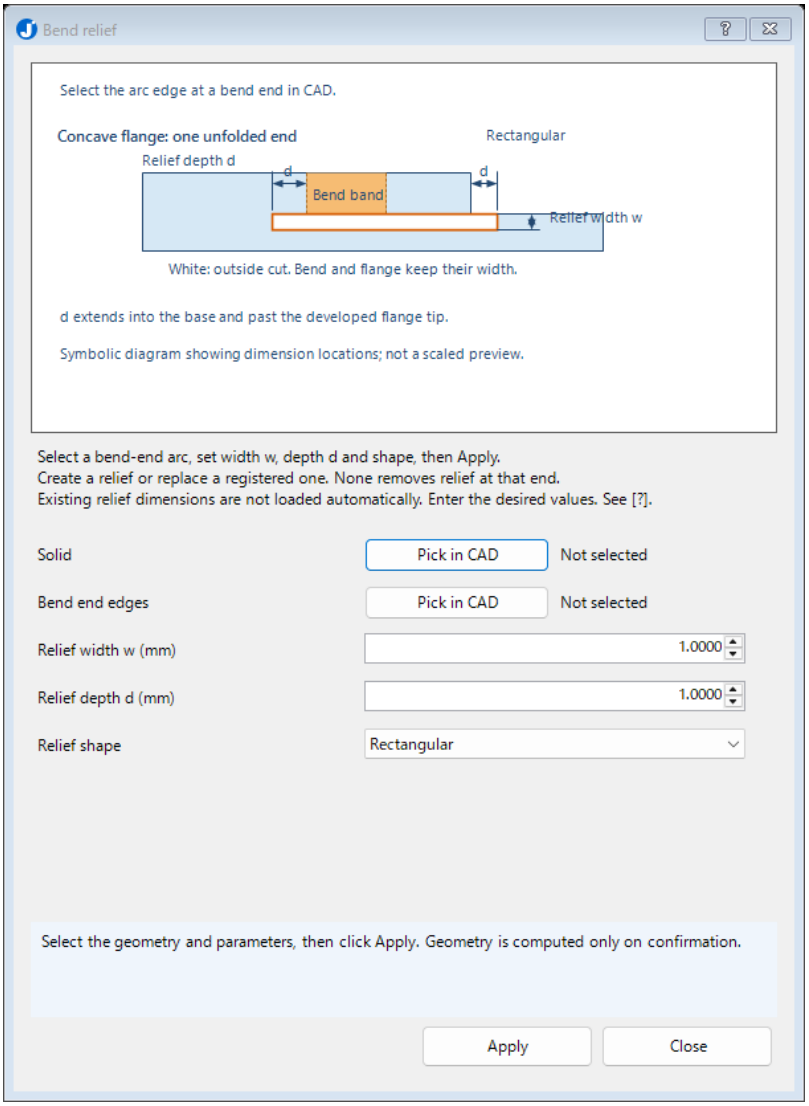


Figure 12 Current dialog. Set the target and inputs, then choose Apply.

Bend allowance table

Allowance is the developed bend length BA (mm), not the deduction from outside dimensions. Choose the scope and operation, then Apply to select a CSV. Selected part retains its blank outline. See the link below for the format and conversion.

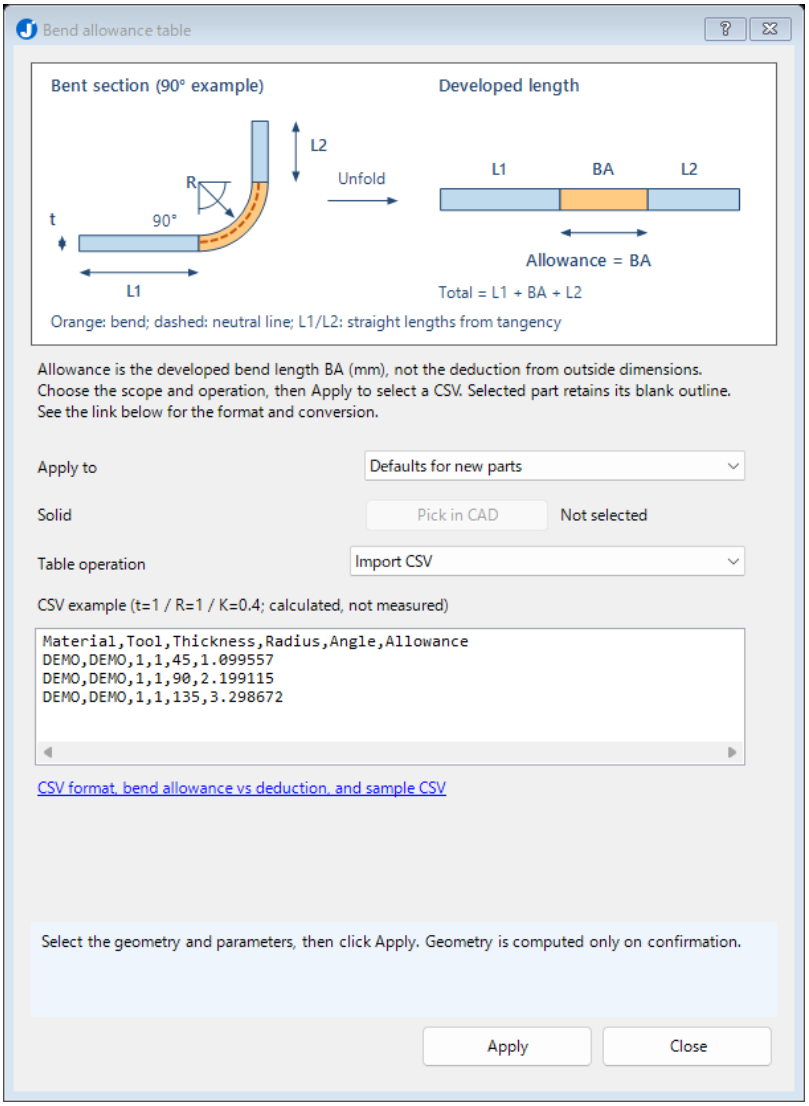


Figure 13 Current dialog. Set the target and inputs, then choose Apply.

Check sheet metal

Check uniform thickness, closed solids, bend connectivity and material overlap in folded/flat states. Manufacturing constraints and machine collisions are outside this check.

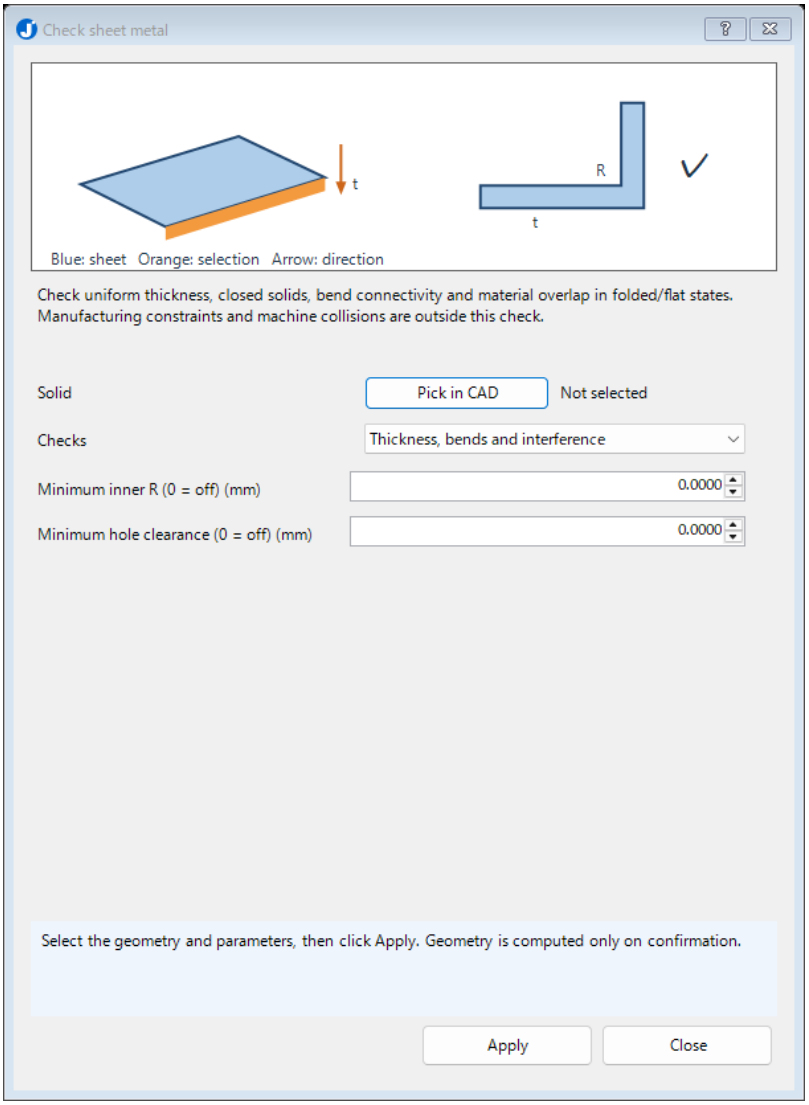


Figure 14 Current dialog. Set the target and inputs, then choose Apply.

Contour flange

Pick one straight edge. Each row defines a relative angle (negative reverses direction), straight length after the bend, and inside radius. Add a row at the bottom; select its header and press Delete to remove it. Lengths are between tangents, not virtual sharp corners.

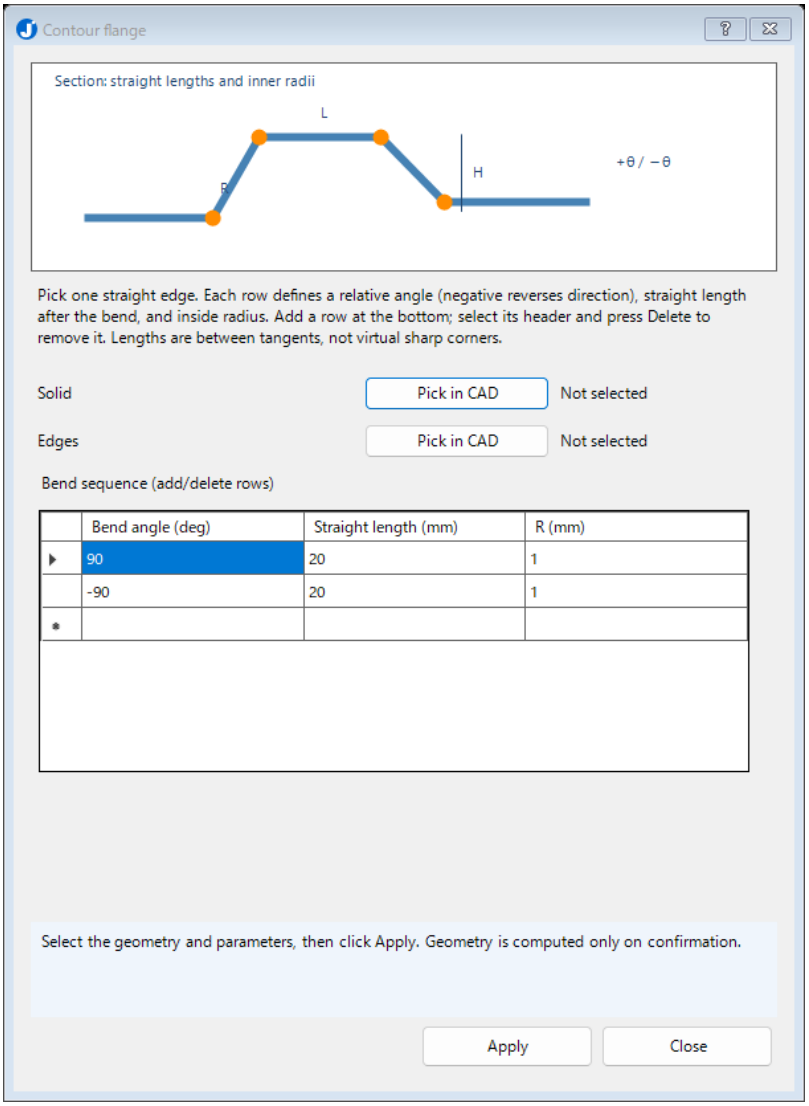


Figure 15 Current dialog. Set the target and inputs, then choose Apply.

Corner treatment

For Miter, select at least two adjoining coplanar flange faces and Apply a seam gap.
Both inward and outward rims are supported, before individual machining. Butt retains the original end-setback operation.

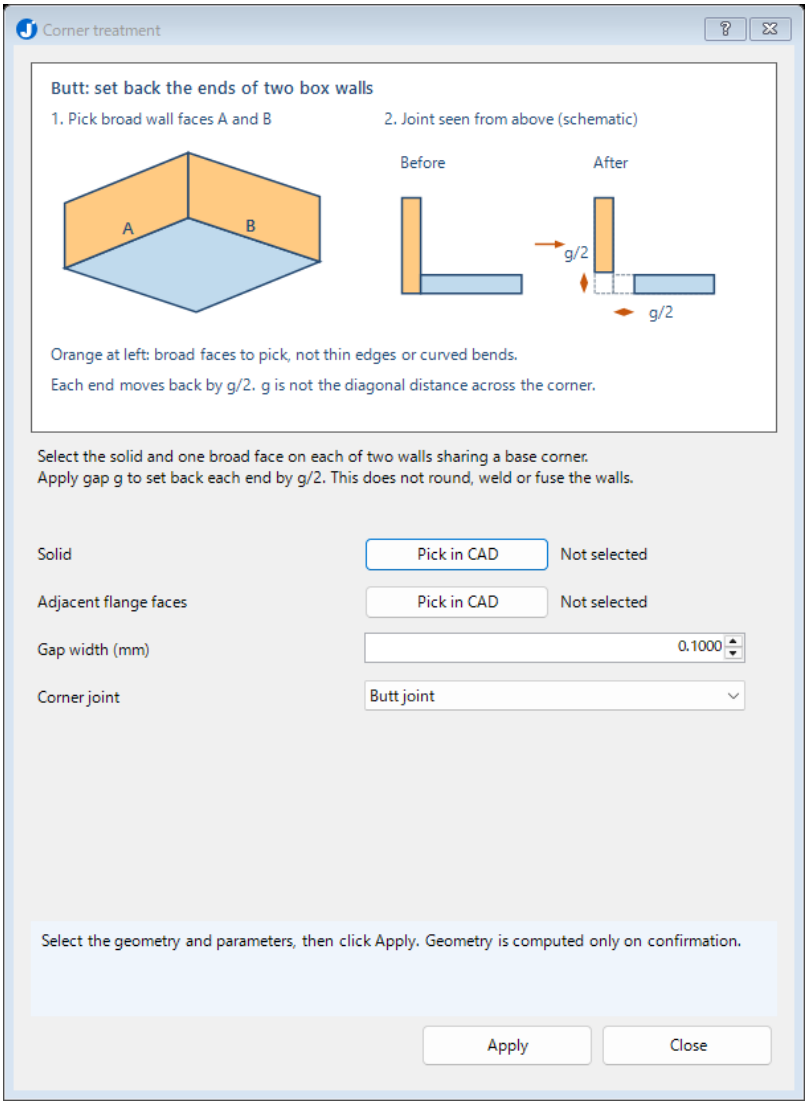


Figure 16 Current dialog. Set the target and inputs, then choose Apply.

Edge flange

Pick a straight edge and select alignment and length reference. Bend start / Straight after bend keeps the previous definition.
Enable automatic miters for adjoining box rims. Gap is perpendicular to the seam.

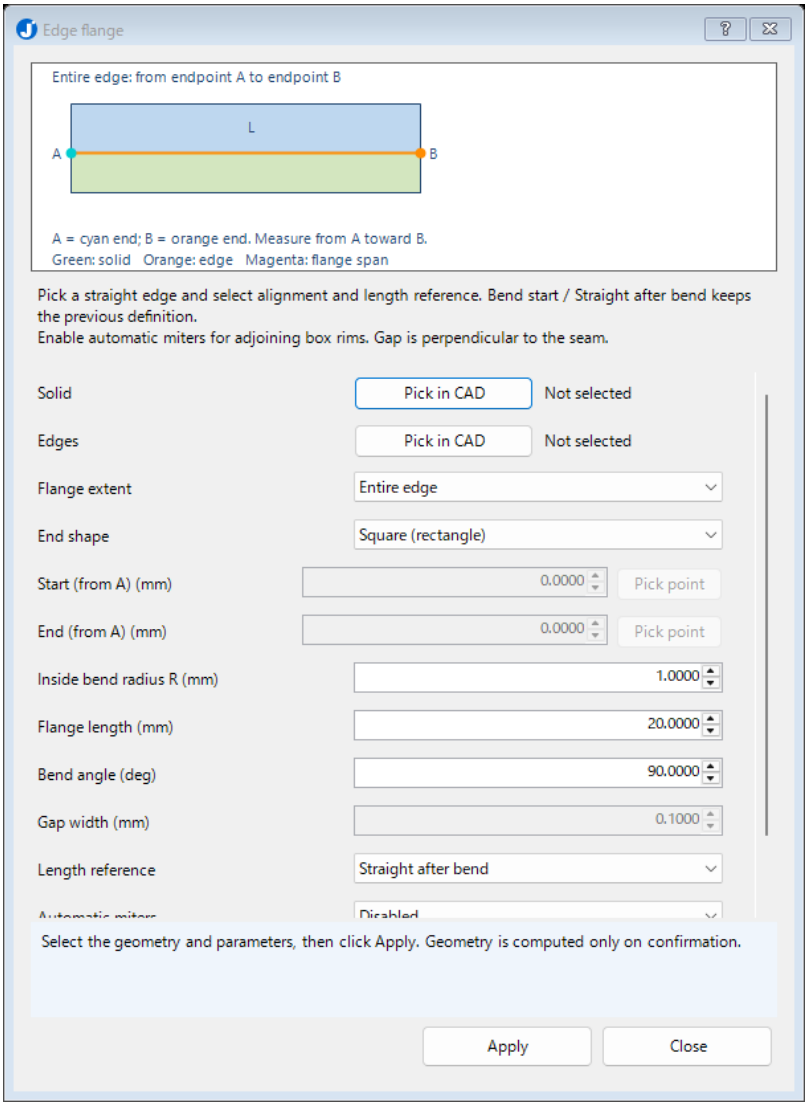


Figure 17 Current dialog. Set the target and inputs, then choose Apply.

Edit bend / flange

Pick the solid and cylindrical bend face to edit angle and inside radius.
Holes, slots and notches across bends are retained. Machined and line-bent blanks keep their developed outline; straight length is calculated.

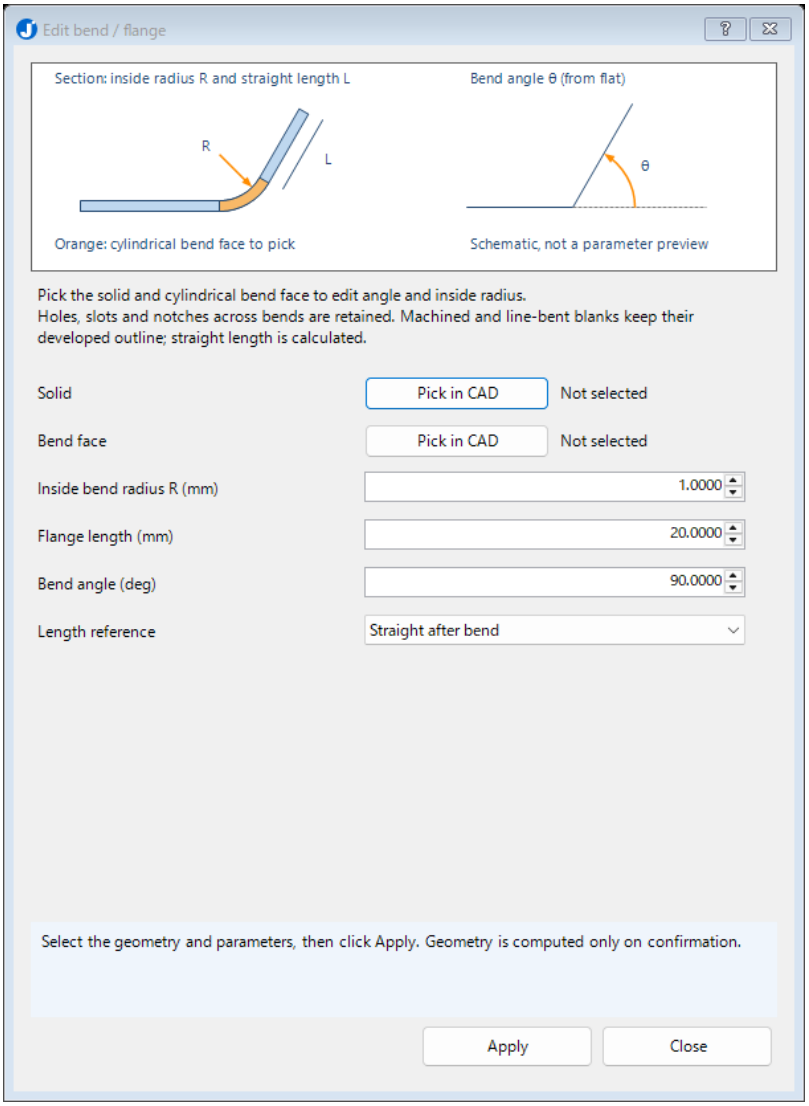


Figure 18 Current dialog. Set the target and inputs, then choose Apply.

Export flat pattern

Save flat outlines, holes and bend centre lines as DXF or DWG.
Choose the output units. The drawing and source solid are retained.

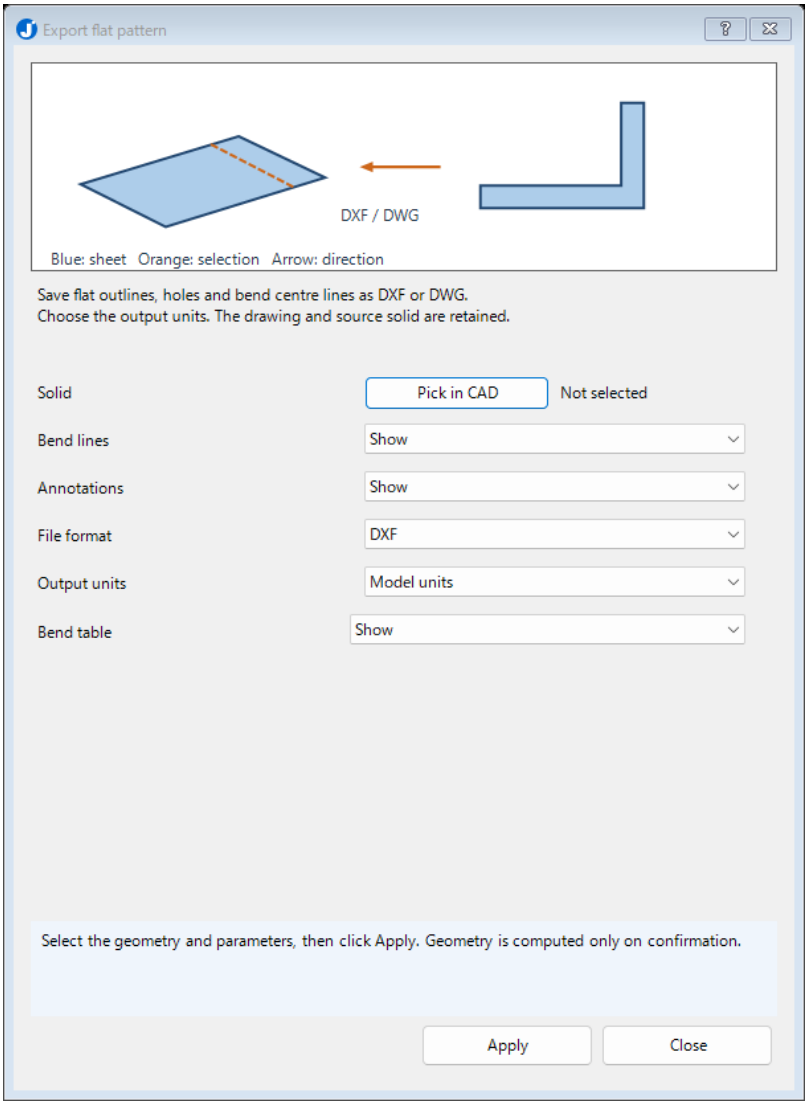


Figure 19 Current dialog. Set the target and inputs, then choose Apply.

Create / update flat pattern

Place outlines, holes, bend centre lines and notes on the model XY plane.
Applying again replaces this part's previous flat pattern and retains the original solid.

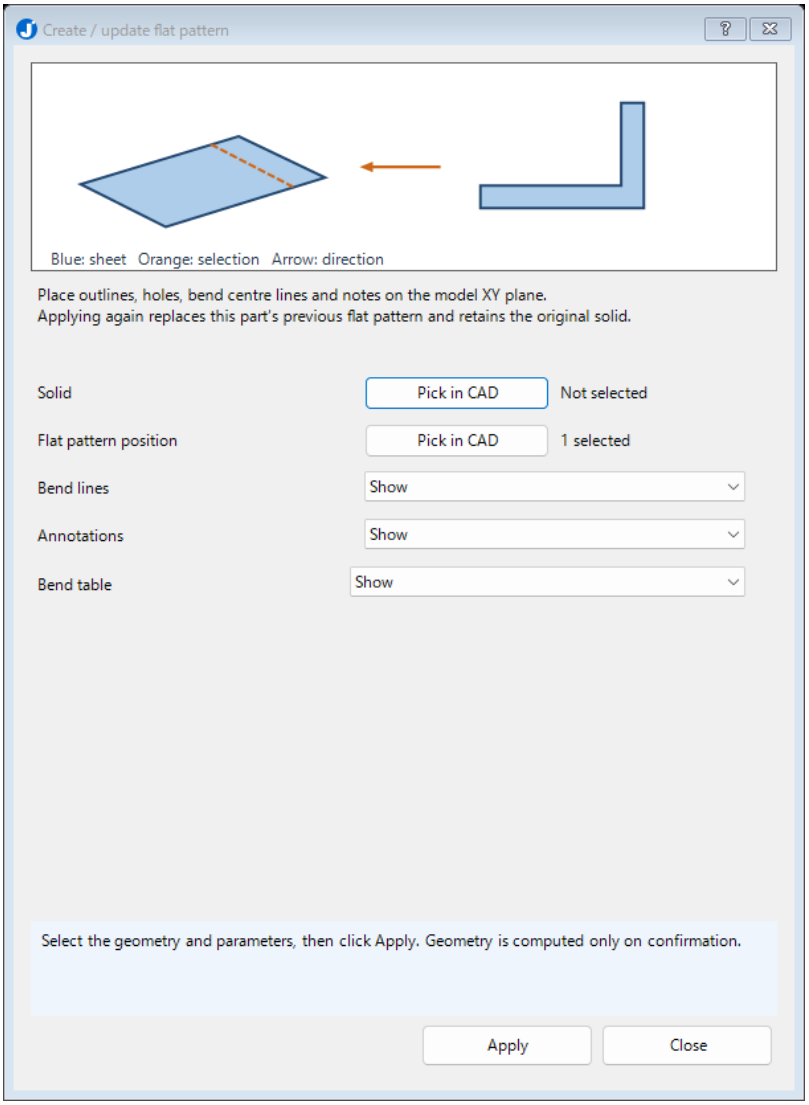


Figure 20 Current dialog. Set the target and inputs, then choose Apply.

Sheet metal forming

Pick a solid, planar face and center, set dimensions and Apply. Unfold retains the burring pilot; louvers become flat with a shear slit in the flat pattern. F numbers follow creation order. Shoulder R must exceed thickness; height must accommodate both radii.

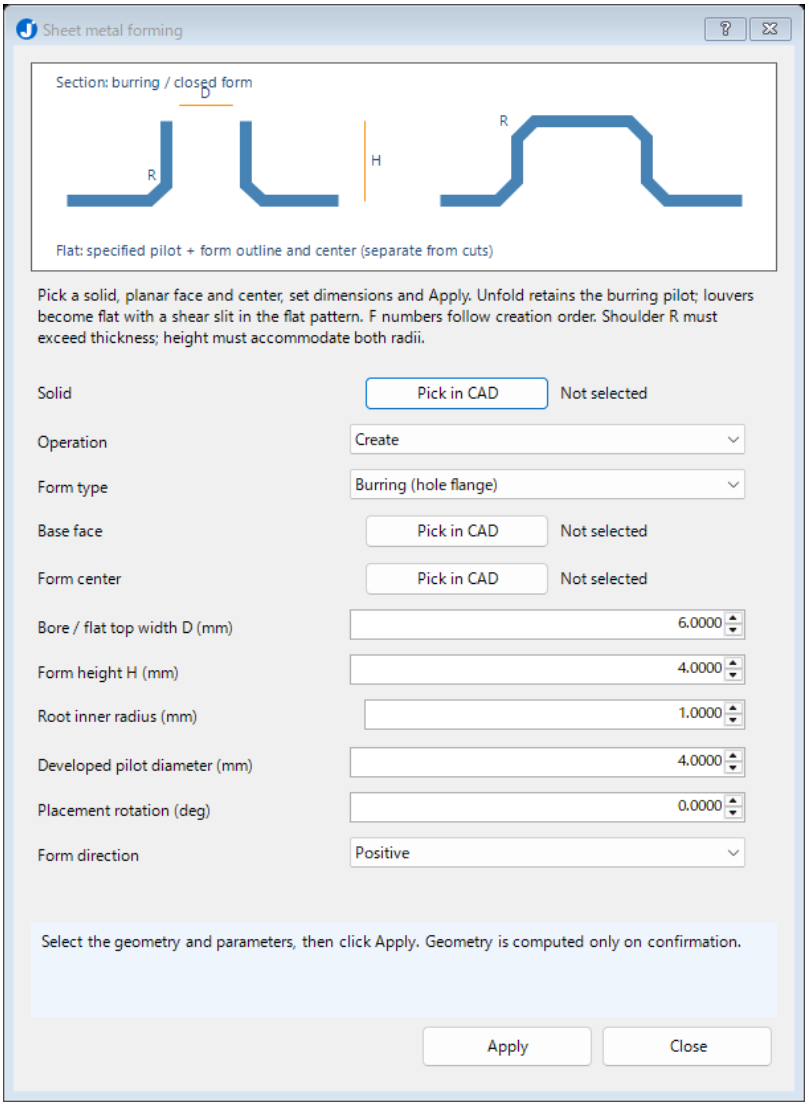


Figure 21 Current dialog. Set the target and inputs, then choose Apply.

Open hem

Pick straight edges, then specify straight return length and the gap between layers. This is a 180-degree open hem with inside R = gap/2; a zero-gap flattened hem is not supported.

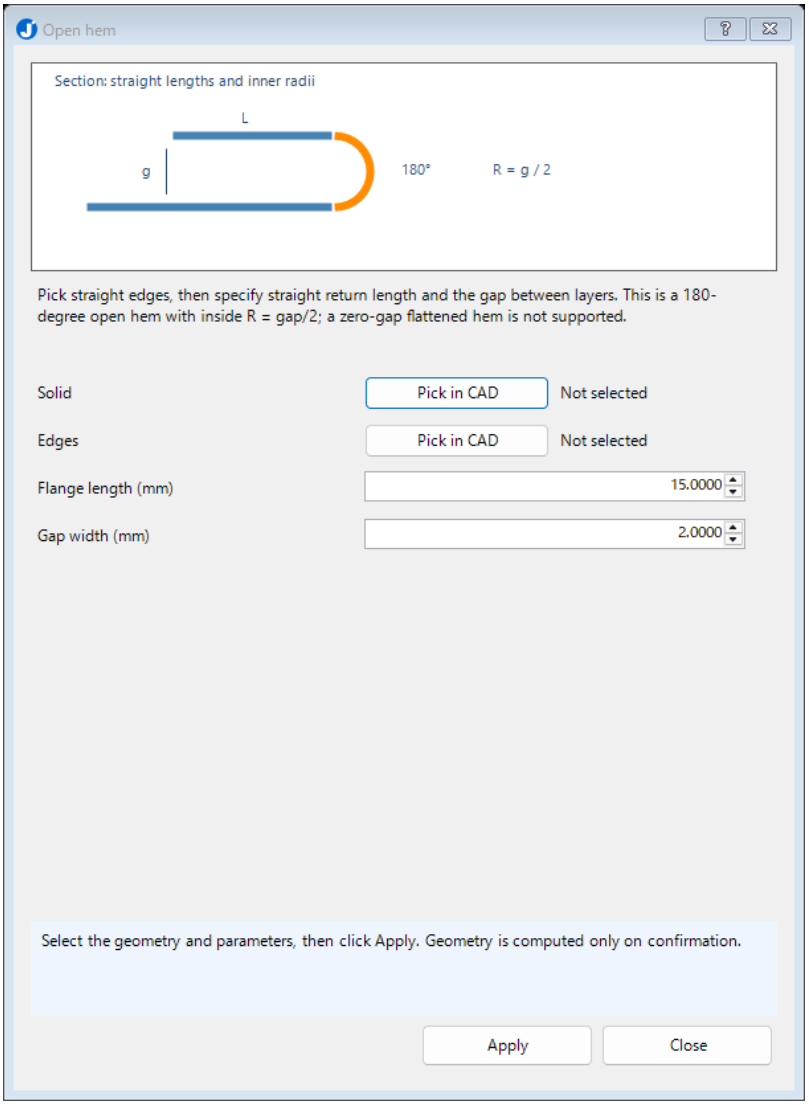


Figure 22 Current dialog. Set the target and inputs, then choose Apply.

Jog

Pick a straight edge and specify step height, final straight length, radius and angle. At 90 degrees, height must exceed 2R + thickness. The second bend reverses direction.

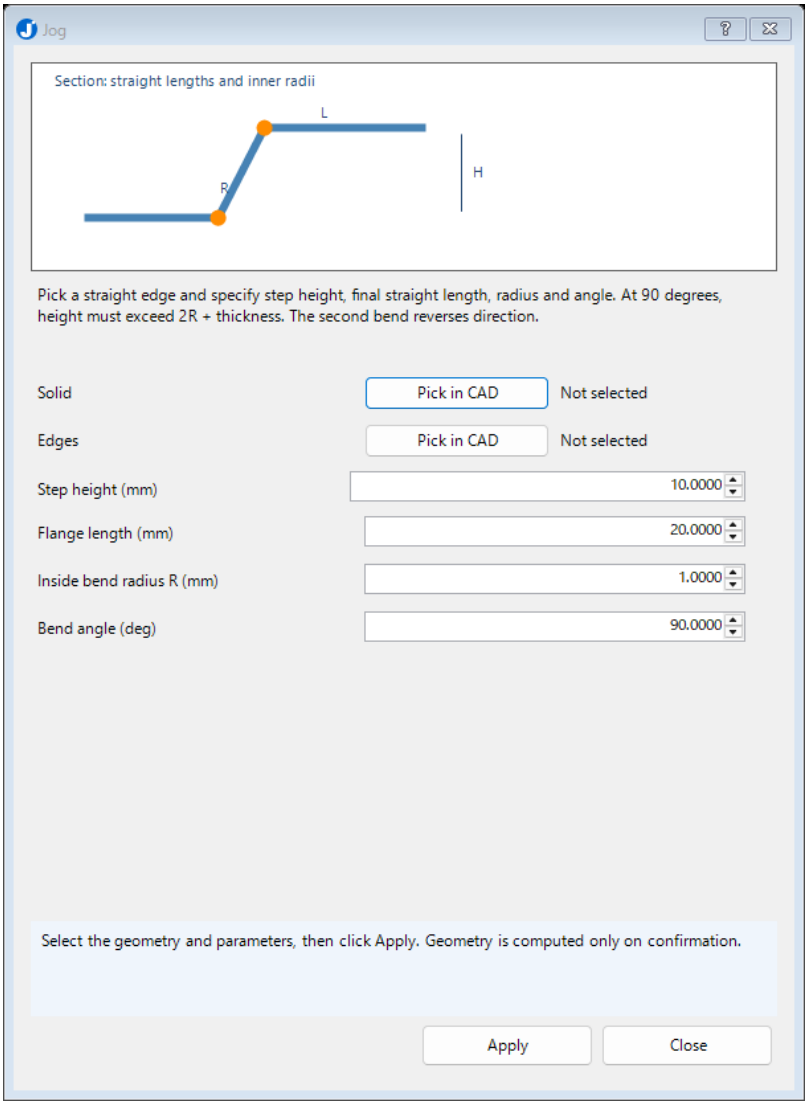


Figure 23 Current dialog. Set the target and inputs, then choose Apply.

Convert to sheet metal

Register a uniform flat plate as sheet metal. Select the whole solid and a broad reference face.
Thickness is detected on Apply. Through holes are retained; existing bent parts are not supported.

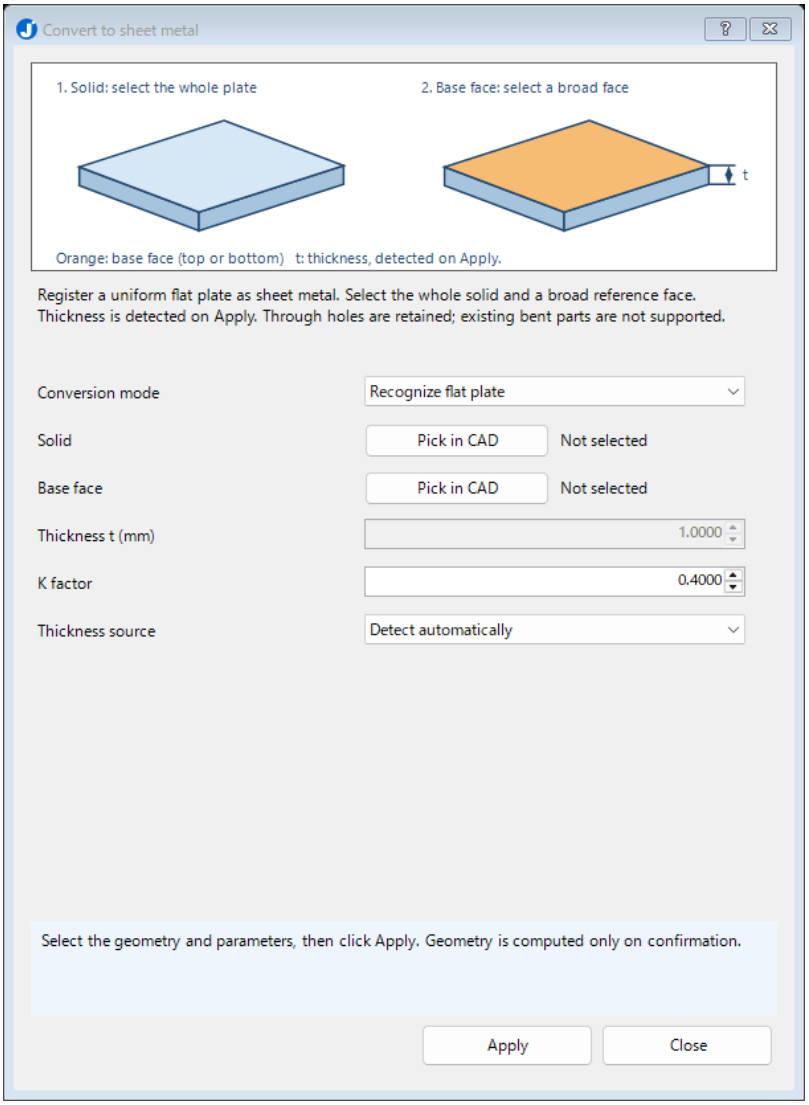


Figure 24 Current dialog. Set the target and inputs, then choose Apply.

Refold

Refold with outline edits and through cuts, including slots across bends.
The sheet must keep its thickness and connectivity. Blind pockets, thickness changes and collisions are rejected.

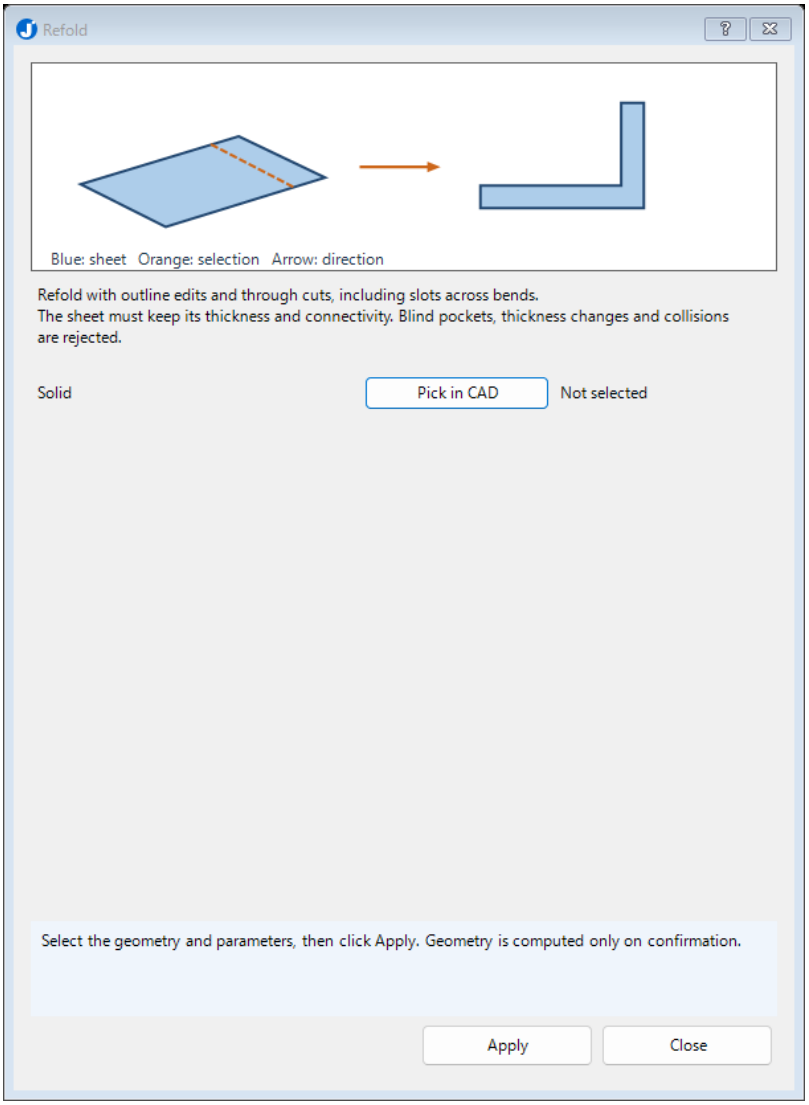


Figure 25 Current dialog. Set the target and inputs, then choose Apply.

Create rip

Open a box corner or cut a through slit along a line on a planar face. Choose a type, select the solid and references, then Apply.

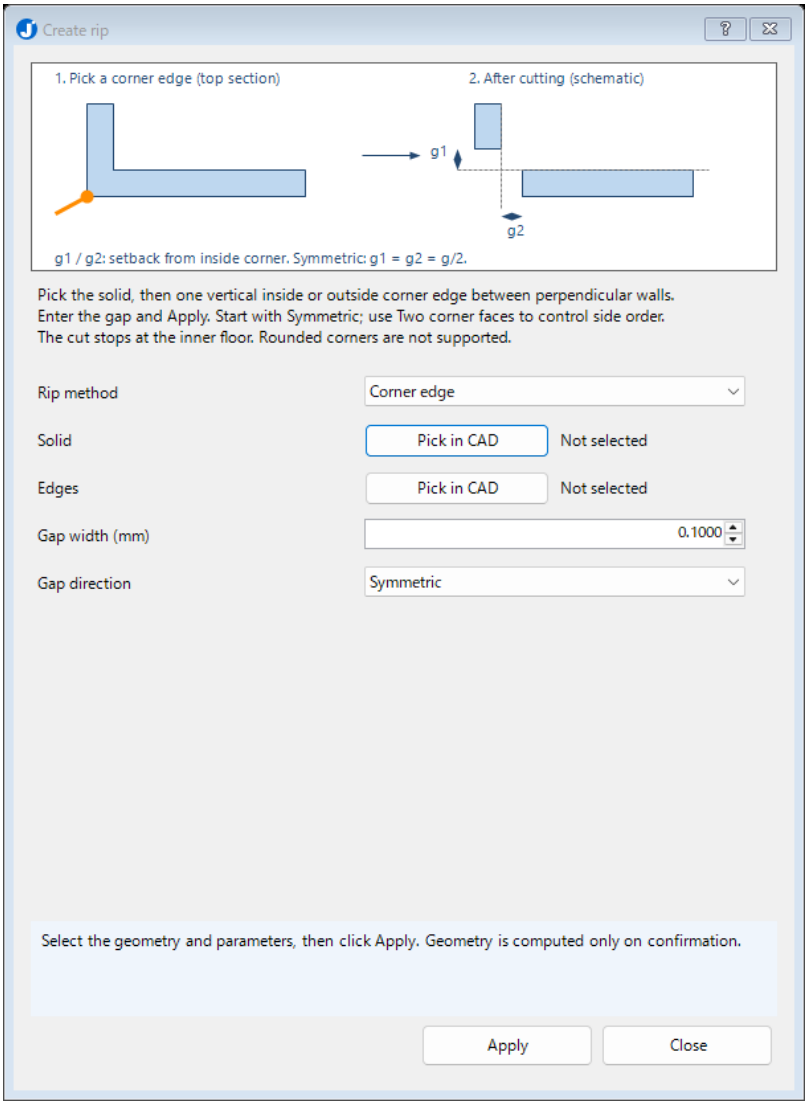


Figure 26 Current dialog. Set the target and inputs, then choose Apply.

Roll / loft

Choose sections and dimensions. Triangulated round–round, rectangle–round and rectangle–rectangle transitions. See [?].

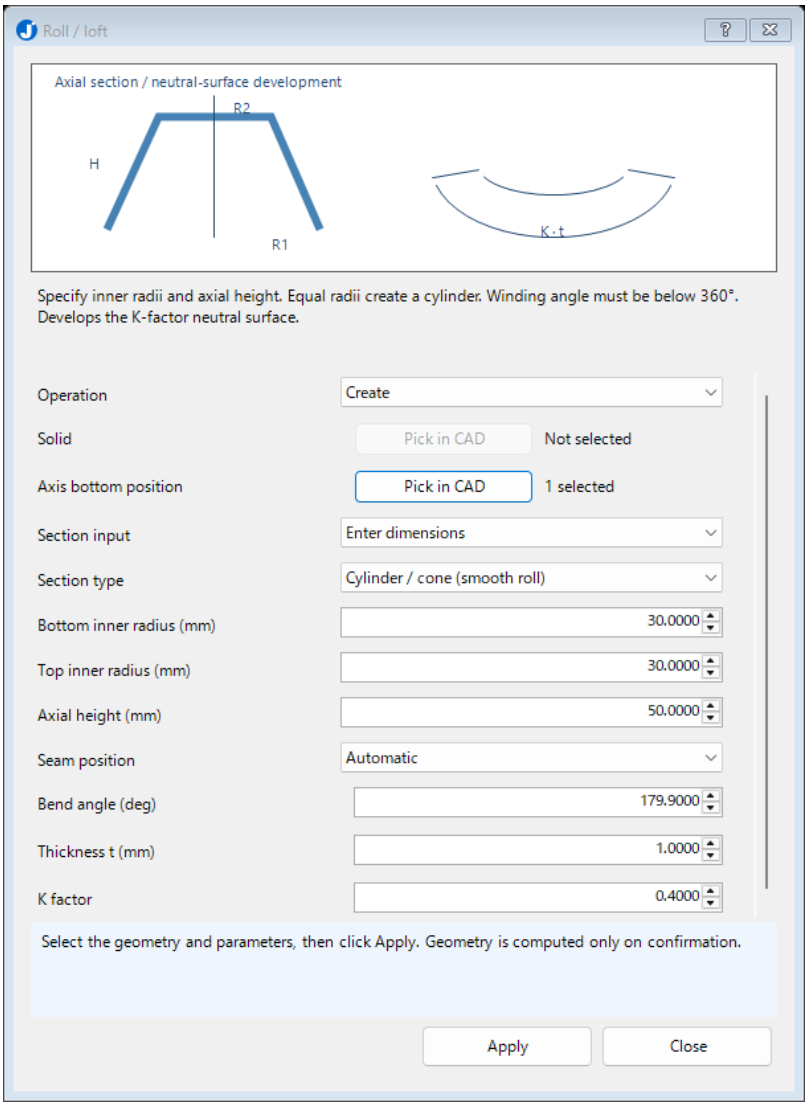


Figure 27 Current dialog. Set the target and inputs, then choose Apply.

Sheet metal settings

For new-part defaults, no solid selection is needed.
Under Apply to, choose Selected sheet metal part to edit an existing part's settings.
Saving does not alter geometry. New flanges use relief settings at re-entrant ends.

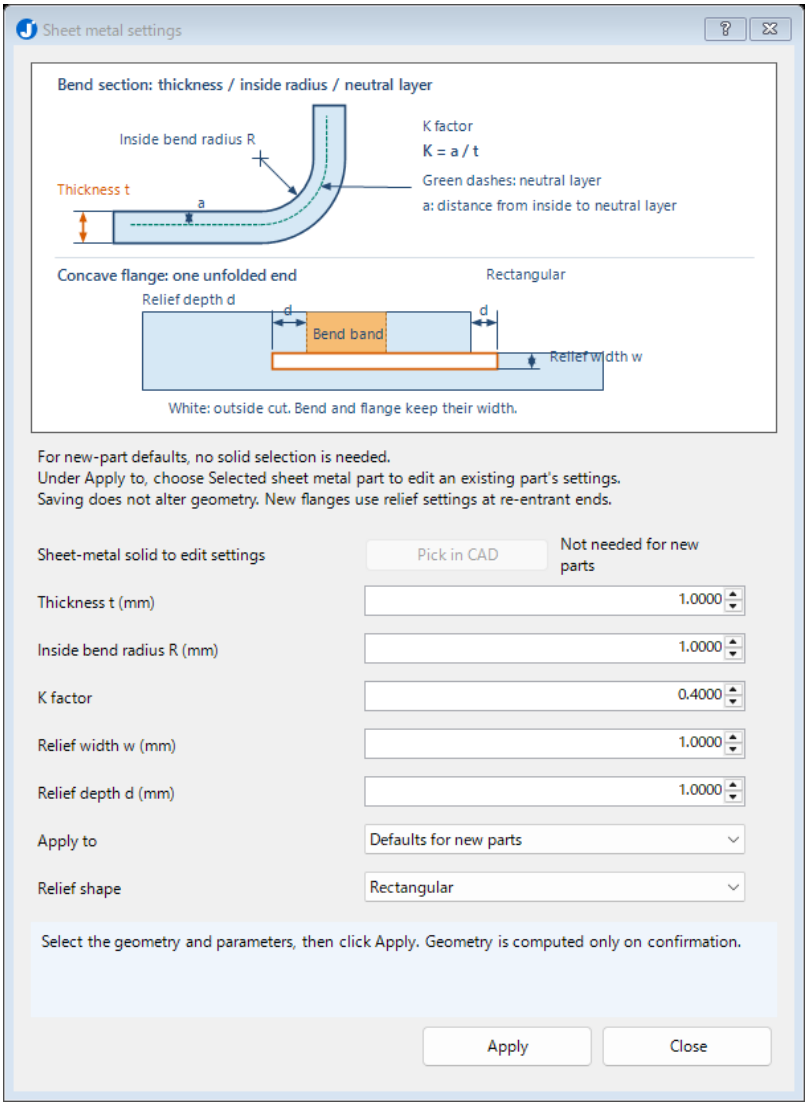


Figure 28 Current dialog. Set the target and inputs, then choose Apply.

Unfold

Unfold with the base fixed. Cyan bend-band center lines support draft references and endpoint, midpoint and on-line snaps.
Edit the outline or make through holes and slots across bends with regular commands, then refold.

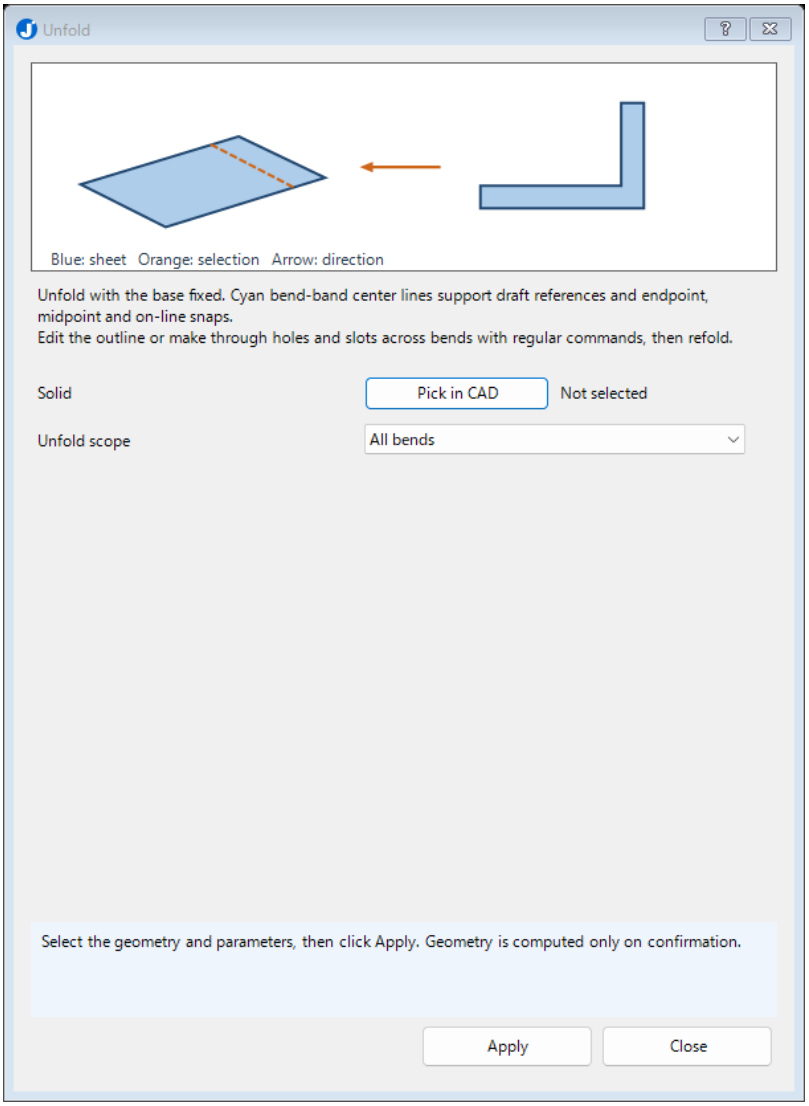


Figure 29 Current dialog. Set the target and inputs, then choose Apply.

Sheet metal settings: illustrated dimensions

Next: Convert to sheet metal (solid, base face and thickness)Create from a profile: Base plate (thickness and direction)

Create rip: box corners and planar slits

Convert an open box: bottom, bend radius and corner clearance

Regular face commands: widths, slots and trapezoidal flanges

Edge flanges: existing panels and partial spans

Bend relief: create, edit and remove

1. Start with Sheet metal settings

These settings store conditions for new parts; applying them does not create a plate or a relief.

• Select Defaults for new parts first. No solid selection is required.

• Enter thickness t, inside bend radius R, K factor, relief width w, depth d and shape.

• Select Apply, then Close, and save the drawing.

• Create a Base plate from a closed profile, then add an Edge flange or a line bend. Use Convert to sheet metal for an existing uniform flat solid.

• A new Edge flange automatically uses the part's relief settings at re-entrant ends adjoining the base sheet. Free ends stay unchanged; None creates no notch.

To add or change relief on an existing flange, or at a free end, use Bend relief and select its end arc. Opening an old drawing does not add reliefs. To change settings for the next flange on an existing part, edit Selected sheet metal part rather than Defaults for new parts.

Sheet-metal solid to edit settings identifies the part whose conditions you want to edit. With Defaults for new parts, the status reads Not needed for new parts and the selection button is disabled.

To edit an existing part, choose Selected sheet metal part under Apply to. Follow Select a sheet-metal part, click Pick in CAD, edit the values, and Apply. Both flat parts and parts bent with these commands are supported. For an ordinary flat solid, first use Convert to sheet metal.

Picking a part or changing the scope reloads saved conditions, so edit the values afterwards. Selection alone does not change geometry. Existing thickness cannot be changed here; existing bends are not recalculated.

2. Thickness t and inside bend radius R

R measures the inner arc. The outer radius equals R+t. A smaller R/t makes a tighter inner corner; a larger R/t makes a broader curve.

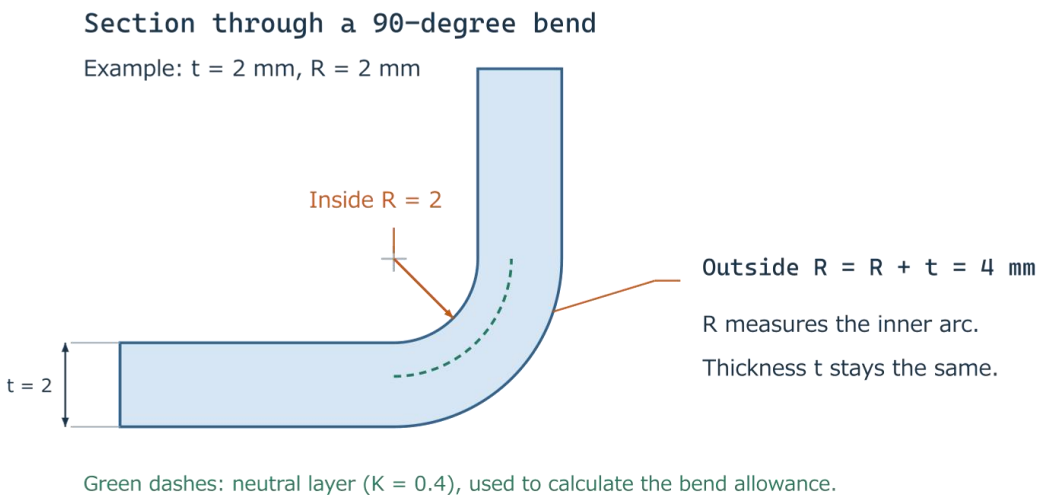


Figure 30 Section of a 2 mm sheet with 2 mm inside radius and 4 mm outside radius.

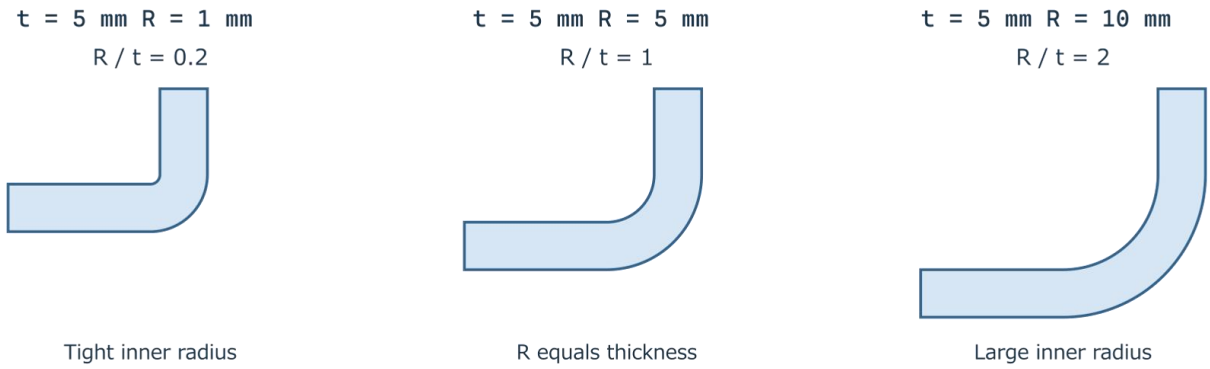


Figure 31 At the same scale, a 5 mm sheet with inside radii of 1, 5 and 10 mm. R/t is 0.2, 1 and 2.

These are geometric comparisons. The feasible minimum radius depends on material, temper, rolling direction, tooling and bending method. R=t is not a universal manufacturing rule.

Relief outside a concave flange

Keep the same bend and flange width as None, then cut the adjacent stationary sheet outside that width. Rectangular and Round use the same placement; neither narrows only the bend band nor steps the upright at its root.

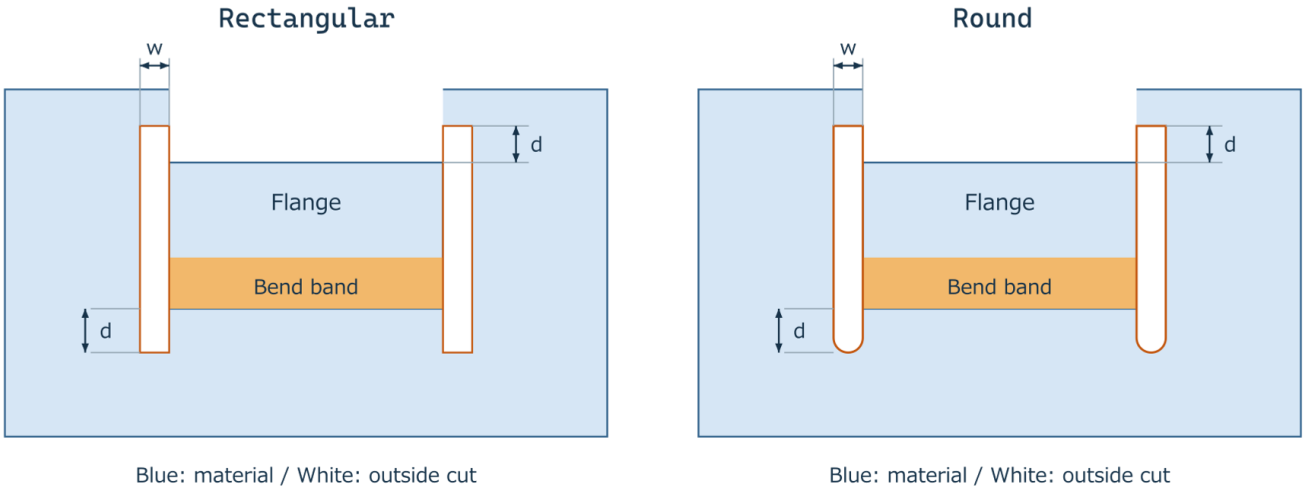


Figure 32 Unfolded U-shaped sheet with full-width flange and bend, and outside slots of width w. Round ends terminate inside the base sheet.

w extends outside the flange. d extends from the bend start into the base. The open end clears the developed flange tip by d so unfolding cannot weld the flange back to the neighboring wings. The cut also covers any greater folded contact reach; existing empty space removes no material.

Round has a closed semicircular end of radius w/2 inside the base, with d measured to its farthest point. Its open end is not capped by a circle tangent to the outside edge. Cuts that disconnect the sheet are rejected.

New flanges use the part settings automatically. Opening or unfolding an older saved relief preserves it. To update, select its end arcs in Bend relief and apply the desired dimensions again.

3. Where width w and depth d are measured

The following diagrams describe the existing free-end / line-bend relief. For outside relief on concave flanges, see Relief outside a concave flange.

w runs inward from the side edge. d extends from each bend-band boundary into the adjacent planar region. The pale orange area is removed through the full thickness. d is not a blind-hole depth.

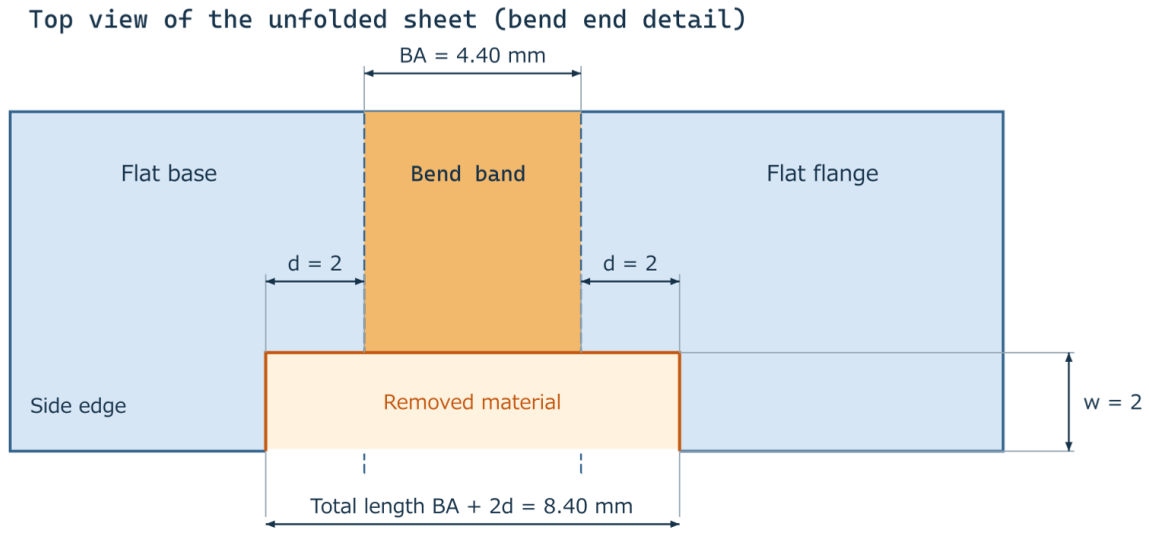


Figure 33 Top view: width w is inward from the sheet side edge; depth d extends from both bend-band boundaries into the base and flange.

JDesignCAD currently applies the same d on both sides, so the full notch length is BA+2d. For a 90-degree bend, BA=pi/2 × (R+K×t).

Example: t=2, R=2, K=0.4, w=2, d=2 mm gives BA=4.40 mm and a full notch length of 8.40 mm. Increasing w retreats the bend end further inward; increasing d lengthens the notch into both planar regions.

This measurement convention describes JDesignCAD. Check reference points and one-sided versus two-sided depth before copying a value from another CAD system.

4. Rectangular and rounded-corner reliefs

The following diagrams describe the existing free-end / line-bend relief. For outside relief on concave flanges, see Relief outside a concave flange.

Round reliefs now have rounded inner corners and a straight mouth opening through the sheet edge. The previous circular-cap construction was corrected because it left thin pointed lips at the outer-edge tangencies. The diagram shows the corrected profile.

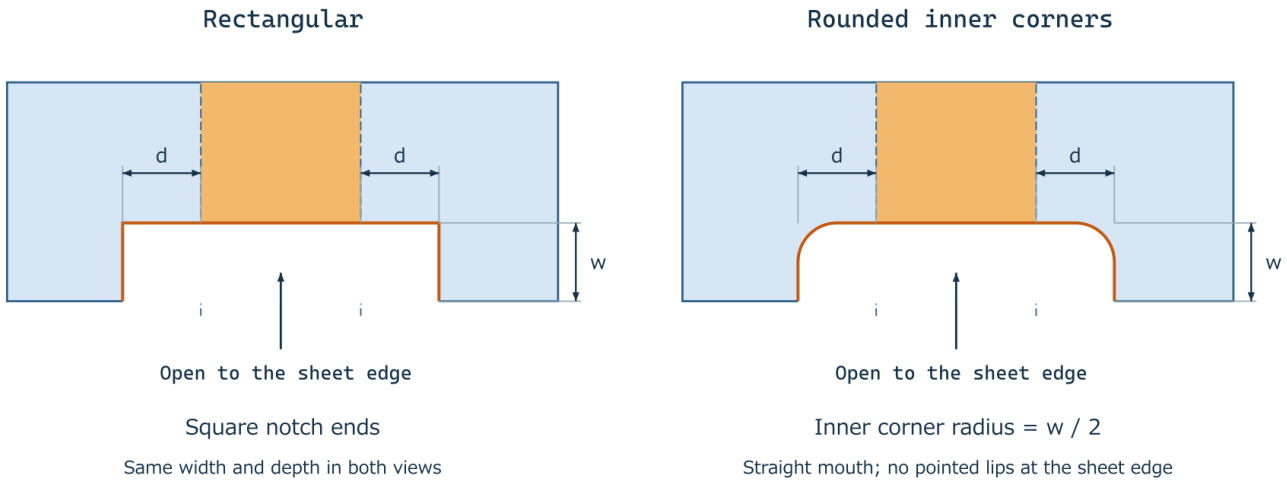


Figure 34 Rectangular and corrected round reliefs. Quarter-circle inner corners have radius w/2; straight sides open through the outer edge.

Rectangular reliefs have square inner corners; round reliefs use quarter-circle inner corners of radius w/2. This radius is separate from bend radius R. d measures from the bend-band boundary to the furthest extent of the notch; do not add the corner radius again.

The white area is an open notch, not a closed obround hole. Think of rounding the inner corners of a notch. The radius is determined by w/2; there is no independent relief-corner radius setting.

Opening an older drawing does not replace its stored relief geometry. Unfold and refold the part, or reapply Bend relief with the same dimensions, to rebuild the corrected profile.

This differs from rounding a sheet-outline corner, or making a circular corner relief where two bends meet. This diagram describes relief at the end of a single bend.

Current geometry limits: w must be smaller than half the bend-axis length, and d smaller than the straight flange length. Round reliefs additionally require d > w/2. For example, w=2 mm and d=1 mm is rejected; d=2 mm meets that condition. Dimensions very close to the radius limit may still fail the solid geometry checks. These checks do not certify manufacturability.

5. Dialog diagrams and None

The diagrams in Settings and Bend relief identify the input fields: thickness t, inside bend radius R, K factor, relief width w and relief depth d. They contain no example dimensions and are symbolic, not previews scaled to the current values. Focusing a numeric input highlights its dimension in orange. Green dashes mark the neutral layer; a is its distance from the inside surface, and K=a/t.

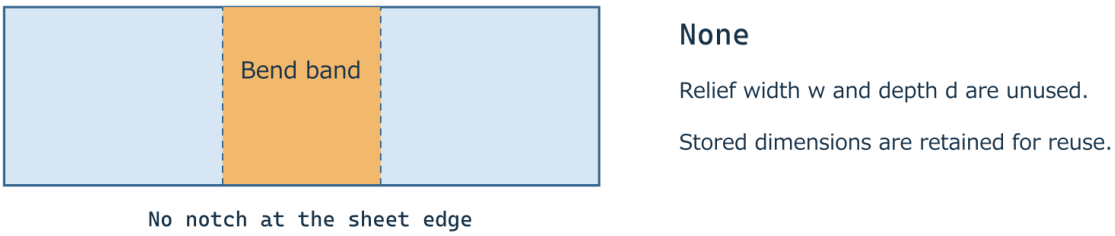


Figure 35 None: no notch at the bend end; width w and depth d are unused.

Select None in Relief shape when no notch is needed. Width and depth are disabled; do not enter zero. Their saved values are retained and become editable again when Rectangular or Round is selected.

• Choose the target in Settings, then None → Apply. Existing geometry is unchanged; the next Edge flange created with those settings receives no automatic relief.

• To remove existing relief, choose Bend relief, select the part and an arc at its bend end, then choose None → Apply. Only the selected end is restored; select both ends to remove both reliefs.

• This removes reliefs created by these sheet-metal commands. It does not fill arbitrary holes, notches or corner-treatment gaps.

None specifies a shape without a relief notch. It does not simulate tearing or plastic deformation during manufacturing.

6. Reading values relative to thickness

The following diagrams describe the existing free-end / line-bend relief. For outside relief on concave flanges, see Relief outside a concave flange.

These examples use R=w=d=t, K=0.4 and a 90-degree bend to illustrate proportional geometry. They are not recommended manufacturing settings or minimum dimensions.

Thickness t	Inside R	K	Width w	Depth d	90° BA
1 mm	1 mm	0.4	1 mm	1 mm	2.20 mm
2 mm	2 mm	0.4	2 mm	2 mm	4.40 mm
3 mm	3 mm	0.4	3 mm	3 mm	6.60 mm
5 mm	5 mm	0.4	5 mm	5 mm	11.00 mm

The shown settings t=5, R=1, w=1 and d=1 mm correspond to R/t=w/t=d/t=0.2: a 5 mm sheet with a 1 mm inside radius and small 1 mm reliefs. Outside R is 6 mm; BA is 4.71 mm; full notch length is 6.71 mm.

K cannot be determined from thickness alone. Compare geometry with a fixed K; use measured results or the fabricator's bend tables for production flat patterns.

7. Materials and the next operations

Representative materials include SPCC cold-rolled steel, SECC electrogalvanized steel, SUS304 stainless steel and A5052 aluminium alloy. Material name alone does not determine R or K; consider thickness, temper and the process. The current settings dialog does not automatically select bending conditions from a material choice.

Settings → Base plate / Convert to sheet metal → Create bends → Bend relief as needed → Unfold → Through cuts within planar regions → Refold

Unfold/refold currently supports bends created with these sheet-metal commands. Cuts across bend bands, blind holes, added material and thickness changes are not supported.

References

JDesignCAD-specific w/d conventions were checked against SheetMetalCuts.cs, BA against SheetMetalBend.cs, and setting scope against SheetMetalCommandAction.cs. The sources below explain general terms and process dependence. Diagrams were created for this help.

• Autodesk Fusion: Sheet metal rule reference

• Autodesk Inventor: About Bend Tables for Sheet Metal Materials

• Protolabs: Sheet Metal Bend Radius Guidelines

Convert to sheet metal: selection and workflow

Sharp box conversion also accepts unequal wall heights. Connected box-width moves support angled walls when bend axes follow the floor sides; forms on the translated wall follow it. Stretching regions must remain one interval along the motion direction. Holes keep their size. Compatible planar face moves/tapers and partial-unfold edits retain thickness and connectivity. Ordinary mirror preserves sheet metadata and correct skin orientation; scaling is not supported as a sheet-metal edit.

Bend numbers, tables and update checks

Create/update flat pattern and Export independently control bend lines, notes and the bend table. B numbers match the manager. UP/DOWN refers to the developed reference face; BA is bend allowance; WIDTH is bend-axis width. Tables use SM_BEND_TABLE and export to both DXF and DWG.

Sheet-metal Check compares an existing drawing to the current part. If outdated, run Create/update flat pattern. It does not silently redraw the model.

Sheet metal settings help · 日本語 · Printable PDF (Japanese, 4 pages)

1. What this command does

Recognize flat plate registers an existing uniform flat solid as a sheet-metal part. It stays in the same model space and position, retaining its outline, physical thickness and through holes. Apply adds the information used by subsequent sheet-metal operations; it does not bend or resize the plate.

Use it for a flat solid made with regular extrusion tools or imported from another file. A part made with Base plate already has sheet-metal information. For an outline without a solid, use Base plate instead.

2. Select the solid and base face

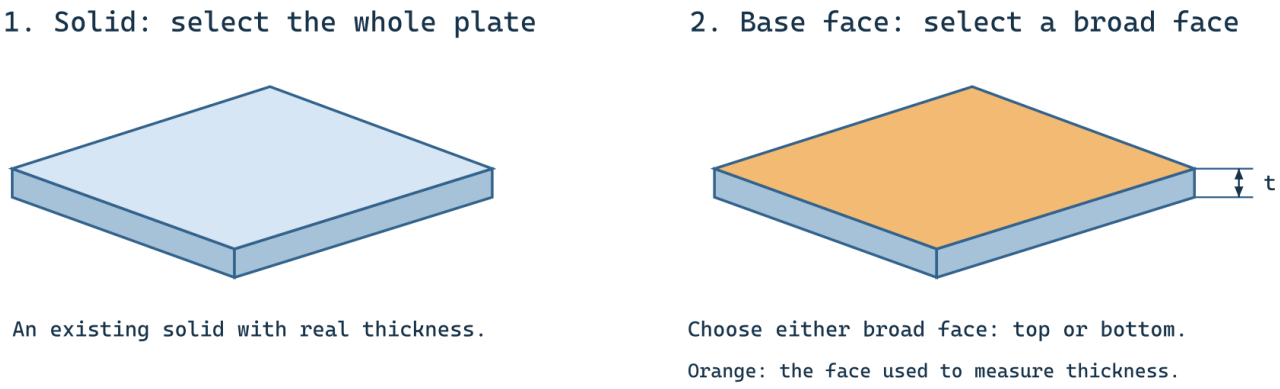


Figure 36 Select the whole plate, then a broad planar face highlighted in orange. t denotes thickness.

• Open Convert to sheet metal.

• Use Pick in CAD beside Solid to select the whole plate.

• Use Pick in CAD beside Base face to select a broad planar face on that same plate. Either the top or bottom face can be used.

The base face is used to measure physical thickness. Avoid narrow side faces, cylindrical hole walls and faces on another part. On a box-shaped solid, a side face can cause another dimension to be interpreted as thickness. This selection does not specify the next bend's position or angle.

Changing the solid clears the previous base-face selection. Select the solid first, then its base face.

3. Thickness source, K factor and Apply

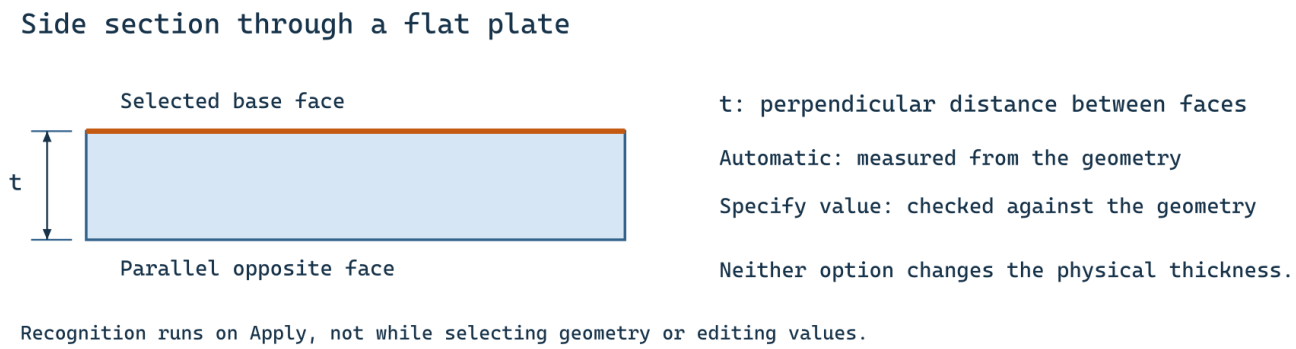


Figure 37 Thickness t is the perpendicular distance between the two broad parallel faces. Automatic measures it; Specify value checks it.

Field	Meaning and use
Thickness source: Detect automatically	The usual choice. The thickness field is disabled. Apply measures the actual geometry. The grey value shown during selection is an initial or loaded condition, not a live measurement.
Thickness source: Specify value	Enter the intended thickness. Apply checks that it matches the geometry. A mismatch is rejected. This does not make the plate thicker or thinner.
K factor	Used for the allowance of subsequent bends. K=a/t, where a is the distance from the future inside surface to the neutral layer. It does not change the flat plate during recognition and is not selected automatically from a material name.



Figure 38 K=a/t: a is the distance from the inside bend surface to the neutral layer within thickness t.

- Normally choose Detect automatically and check the K factor.
- Click Apply. Recognition and geometry checks run at this point.
- After success, click Close and save the drawing.

The registered thickness comes from the geometry. To inspect it, open Sheet metal settings, choose Selected sheet metal part and select the part. Length inputs use the units shown beside the fields. K is dimensionless.

4. Supported and unsupported shapes

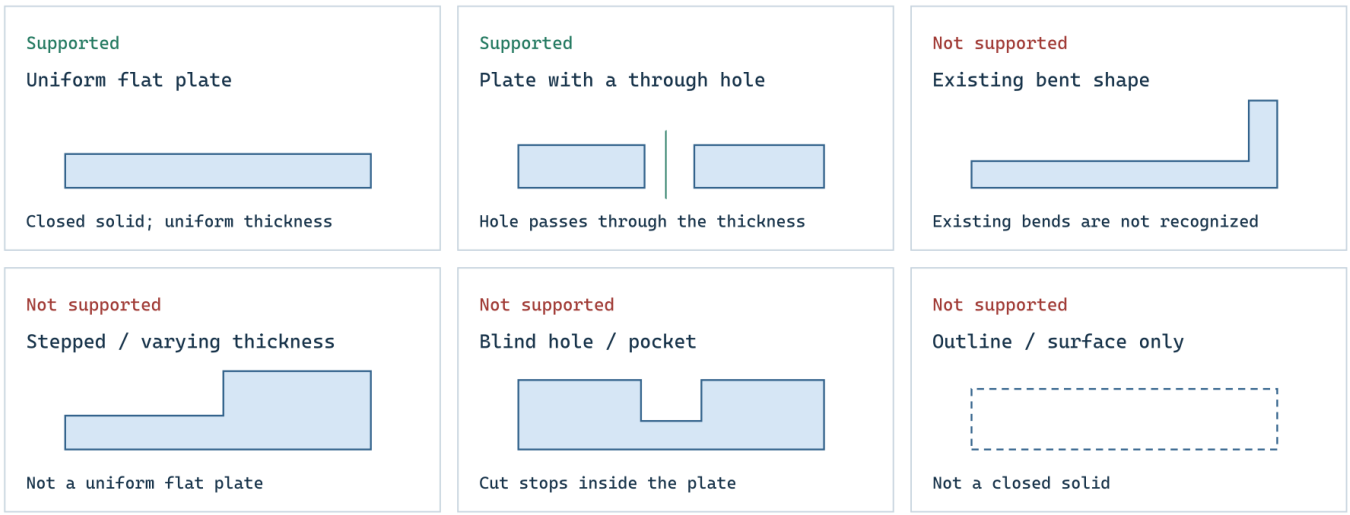


Figure 39 Uniform flat plates, with or without through holes, are supported. Existing bends, steps, blind pockets and outlines without a solid are not.

These are representative section views. The command checks a closed solid, matching opposing faces and boundaries, and uniform thickness. Through holes and through notches are supported when the plate meets those conditions. Steps, taper, or bevels and rounds across the thickness can prevent recognition in this initial implementation.

This is not automatic unfolding of a part already bent in another CAD system. Start with a flat plate, register it, and create bends with JDesignCAD's sheet-metal commands.

Ripping a corner of a box does not convert it into a folded sheet-metal part. Rip cuts the geometry of an ordinary solid. Conversion into unfoldable sheet metal requires a fixed face, bend locations, thickness, inside bend radius and bend allowance, together with the required bend geometry and reliefs. A layout with a fixed bottom and four opening side walls needs rips at all four vertical wall-to-wall corners. The Convert box / L / U-shaped solid mode supports a uniform rectangular bottom with one to four perpendicular walls. Full-height corner seams are supported; holes, steps and existing bend radii remain unsupported. See the illustrated box conversion workflow.

5. Next operations and troubleshooting

Flat solid → Convert to sheet metal → Edge flange / Bend along line → Unfold → Through cuts in planar regions → Refold

Successful recognition does not change the appearance. It registers sheet-metal data in the same model space, without adding another part. Undo reverses the operation. A failed check preserves the original solid.

Symptom	What to check
Cannot select the base face	Select the solid first, then a broad planar face on that same part.
Uniform flat plates only	Check for bends, steps, blind holes, taper or an open surface instead of a closed solid.
Specified thickness does not match	Check the value, displayed unit and base face. Use automatic detection if the thickness is unknown.
Registered thickness is unexpected	Check whether a narrow side face was selected. Also verify the actual dimensions and imported scale.
No visual change after Apply	This is expected: recognition registers information. Create the bend with the next command.

Do not run recognition again on an unfolded sheet-metal part. The current command starts fresh sheet-metal information and does not retain existing bend history. After making through cuts in the unfolded part, use Refold directly.

Base plate: profile, thickness and direction

Sheet metal settings · Convert to sheet metal · 日本語 · Printable PDF (Japanese, 2 pages)

1. What Base plate creates

Add thickness to one closed planar profile to create a new sheet-metal solid. Work stays in model space. The original profile remains in place. Sheet-metal information is assigned automatically, so Convert to sheet metal is not required afterwards.

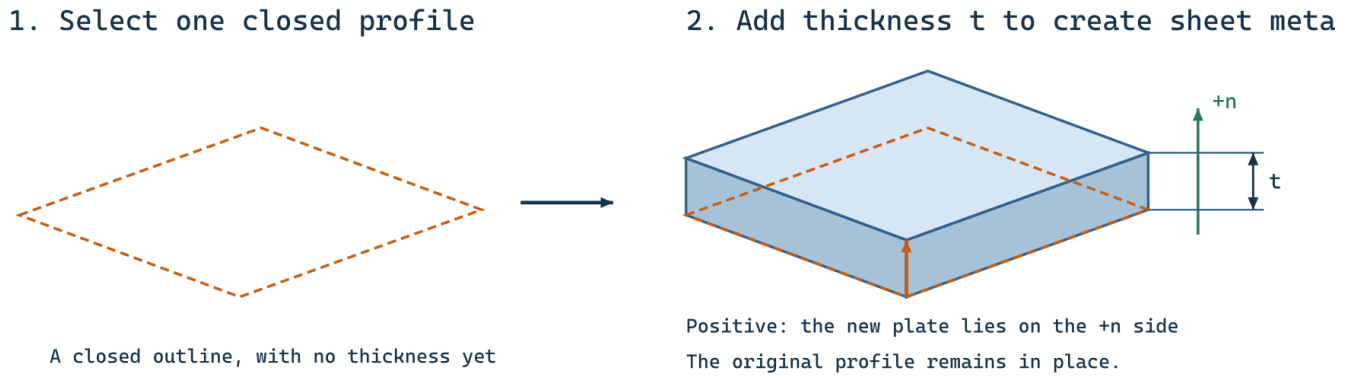


Figure 40 A closed profile becomes a plate of thickness t on the positive normal side, while the source profile stays in place.

2. Workflow

• Prepare one closed planar profile, such as a circle, closed polyline or closed path. Join separate segments into a closed path first.

• Open Sheet metal → Base plate.

• Under Closed profile, click Pick in CAD and select that outline. This field expects a curve object, not an existing solid or its face.

• Enter the total thickness t, using the unit shown by the input field.

• Choose Thickness direction. The schematic above switches to match your choice.

• Click Apply. Check the result, close the dialog and save the drawing.

Selecting a profile or editing values does not create a solid. Creation runs on Apply. Thickness must be positive.

3. Positive, Negative and Symmetric

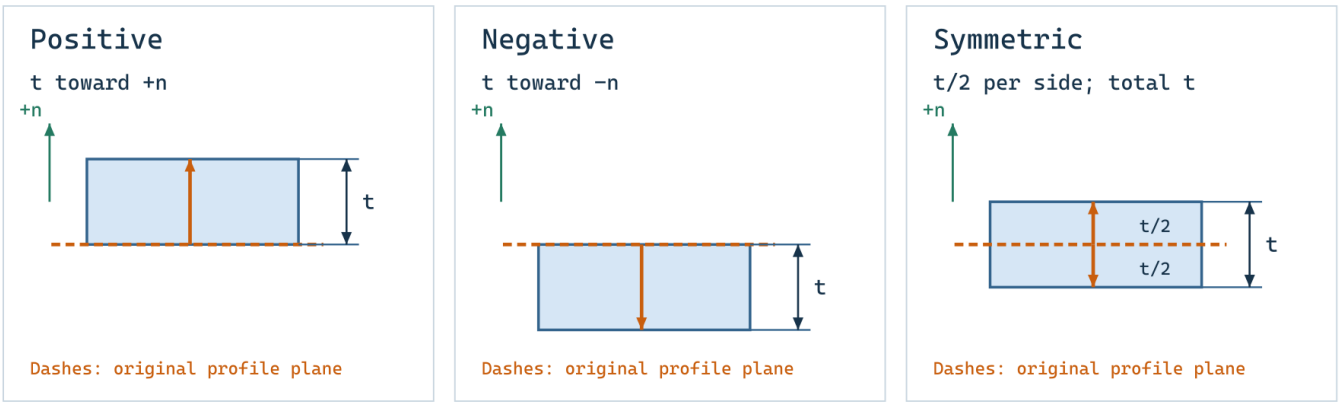


Figure 41 Positive adds t toward positive n, Negative adds t in the opposite direction, and Symmetric adds half the total thickness to each side.

Direction	Position relative to the original profile plane
Positive	Add thickness t toward +n. The profile plane becomes one surface of the plate.
Negative	Add thickness t toward -n, on the opposite side.
Symmetric	Keep the profile plane at mid-thickness and add t/2 to each side. Total thickness is t, not 2t.

On the usual XY work plane, Positive means up (+Z) and Negative means down (-Z). Drawing the profile clockwise or counterclockwise does not change this. Thickness direction is separate from inside/outside bend choices.

+n is perpendicular to the profile plane and oriented toward the +Z side of the work plane captured when the command starts. A tilted work plane uses its own coordinate axes. Rotating the view during the command does not swap the sides. For a profile perpendicular to the work plane, +Y is used to choose the positive side, then +X if needed. The drawing is a schematic, not a preview of the actual profile or entered dimensions.

If the plate is on the wrong side, keep the same dialog open, switch between Positive and Negative and click Apply again. The plate most recently created by this command is replaced with the new result.

4. After creation and troubleshooting

Situation	Action
U-shaped or L-shaped plates	Concave outlines are supported when the entire boundary forms one closed loop. A polyline with coincident start and end points is accepted even when its closed flag is off. Only the working copy is closed; the original outline remains unchanged.
Cannot select the profile	Use one curve object forming a closed loop. Separate lines, outlines with an endpoint gap and solids are not accepted.
Apply fails	Check that the outline is planar and closed, with no self-intersections or invalid overlaps.
A plate with holes is needed	The current command takes one profile, not a simultaneous outer loop and several hole loops. Create the plate first, then make the holes with the usual cutting commands.
Change thickness or direction after creation	While the same dialog stays open, change the values and apply again to replace its previous result. Closing it and starting a new Base plate command creates a new plate. Undo or remove an unwanted earlier plate before starting again.

Closed profile → Base plate → Edge flange / Bend along line → Unfold and cut → Refold

For an existing flat solid, use Convert to sheet metal instead. Creating a base plate does not create bends or reliefs.

Edge flange: base, existing flanges and partial spans

Tool	Inputs
Open hem	180° return; inner radius = gap / 2. L is the returning straight length. Multiple edges are supported. Closed and rolled hems are not included.
Jog	Step height H, final straight length L, inner R and angle. The second bend reverses direction. At 90°, H must exceed 2R + thickness.
Contour flange	Each row specifies incremental angle, straight length and inner radius. A negative angle reverses direction. Add in the last row; select a row header and press Delete to remove it.

At 180°, use straight tangent lengths because the virtual sharp intersection is unbounded.

Bend management

The list shows B number, angle, radius, straight length and parent (BASE = stationary base, OFF = suppressed). Suppress straightens the bend while retaining material. Resume restores it. Delete removes the selected flange and dependent flanges/forms. Parent 0 means the base.

Sheet-metal settings · 日本語

Four top flanges: worked box example

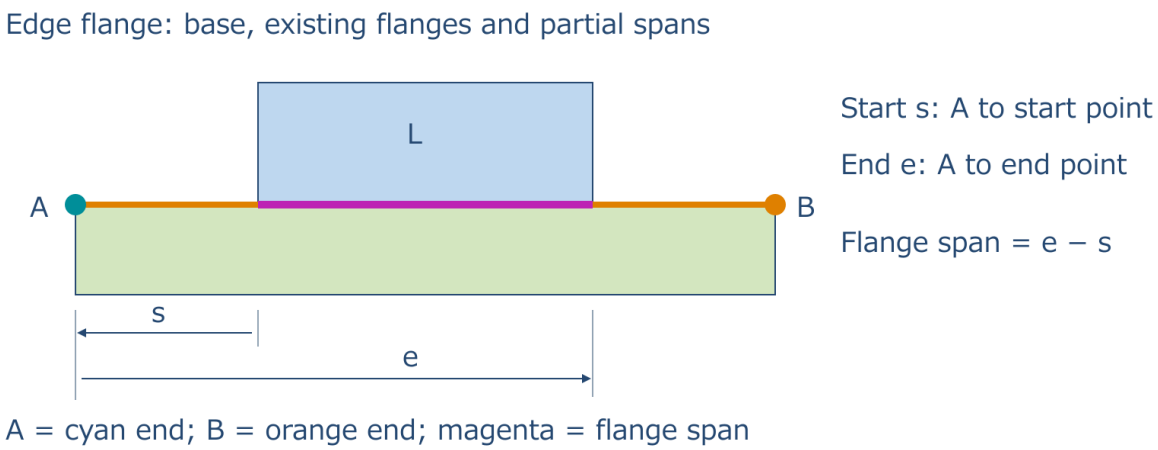


Figure 42 A = cyan end; B = orange end; magenta = flange span

Machined bends, automatic miters and dimension references

Set thickness, inner bend radius and K factor from the drawing or current sheet settings. Retain necessary relief at bend ends and adjoining flanges, while avoiding protrusions and duplicate relief. Overlapping material is not a substitute for a manufacturable connection.

After unfolding and refolding, inspect holes, notches, bend ends and face interference. Ordinary edits can be preserved only for supported thickness directions and bend locations. Thickness changes, blind pockets and unsupported oblique machining across a bend may be rejected. On failure, preserve the original part and review the error before changing the machining approach.

In MCP, sheet.flange accepts startDistance and endDistance for a partial edge. Use layout.create to arrange the flat pattern with dimensions and text leaders, then inspect it with view.capture.

Create a flange on the base or at the tip or side of an existing flange, including a line-bent panel. Further flanges can be added to the result. Use the full edge or a partial span in model space.

Steps

• Open Edge flange and select the sheet-metal solid.

• Select one free straight outer edge of the base plate or an existing flange. The solid is green and the edge is orange.

• Use Entire edge, or choose Start / end positions for a partial span.

• Enter both distances from A, or use each Pick point button on the CAD canvas.

• Set straight length L, bend angle and inside radius; click Apply.

Add to an existing flange

Select the free tip or side of a planar flange. Edges already connected to a bend are not free boundaries. New flanges follow their parent when its angle or radius changes. Changing a regular rectangular parent flange length moves children at its tip; side attachments keep their distance from the tangent. Edits that remove the attachment or cause interference are rejected.

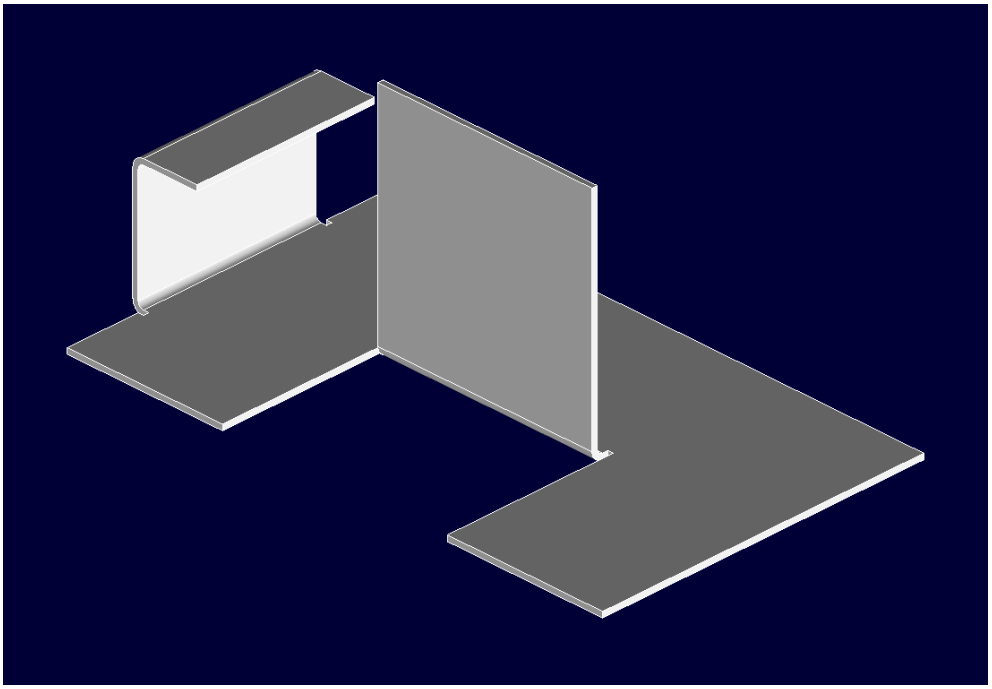


Figure 43 Additional return flange at the tip of an existing upright

Actual result: a return added to the left upright.

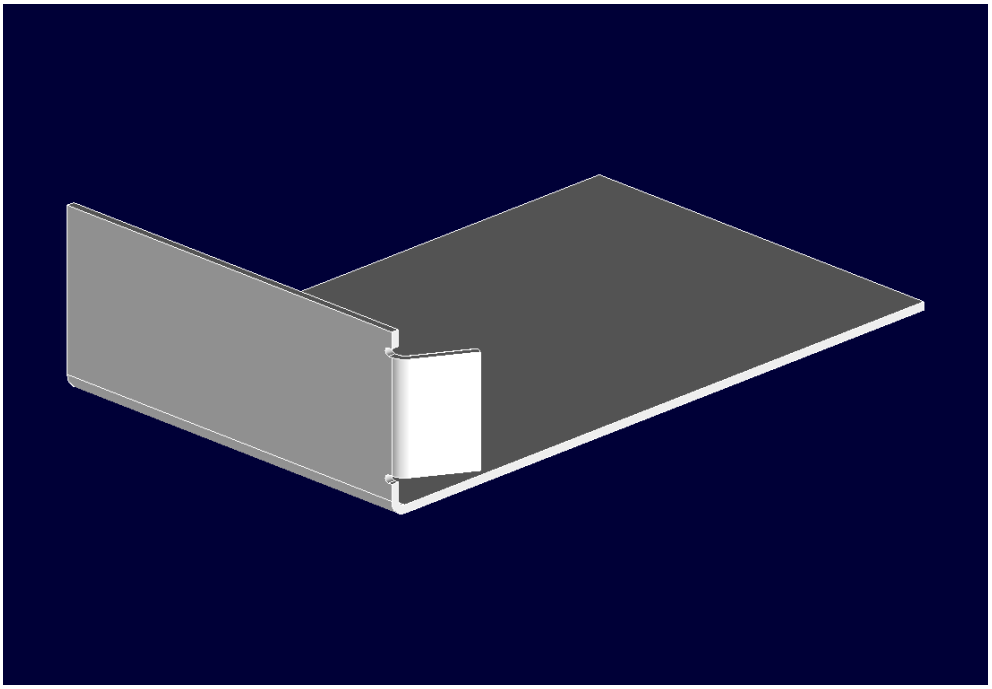


Figure 44 Partial flange on the side of an existing flange

Actual result: a partial side attachment with end reliefs.

Extent and colors

A is the cyan endpoint; B is orange. Both positions are measured from A toward B. End is not a distance from the opposite end. Use 0 ≤ start < end ≤ edge length. The flange span is magenta. The illustration is schematic.

Picked points are projected onto the selected straight edge. Snapping to endpoints or midpoints is supported. Selecting a different edge resets the range to its full length.

Field	Meaning
Start s / End e	The span along the edge; flange width is e − s.
Length L	The straight portion after the bend, distinct from the span along the edge.
Entire edge	Uses A through B; range fields are disabled.

Relief and subsequent operations

Internal range ends use the part's stored rectangular, round or no-relief rule. Free endpoints do not receive additional relief. Geometry is computed on Apply. Undo/Redo, unfold/refold and save/reopen retain the span.

Supported edges belong to the free straight outer boundary of the base or an existing flange. Curved edges, hole edges and existing bend boundaries are not supported. Interfering bends or relief that disconnects the sheet are rejected.

Four top flanges with 45-degree ends

A new rectangular flange overlaps its neighbor before it can be tapered. Select End shape → Both ends 45 deg - narrow tip to create a trapezoid directly. Use a partial range to keep adjacent bend bands clear as well.

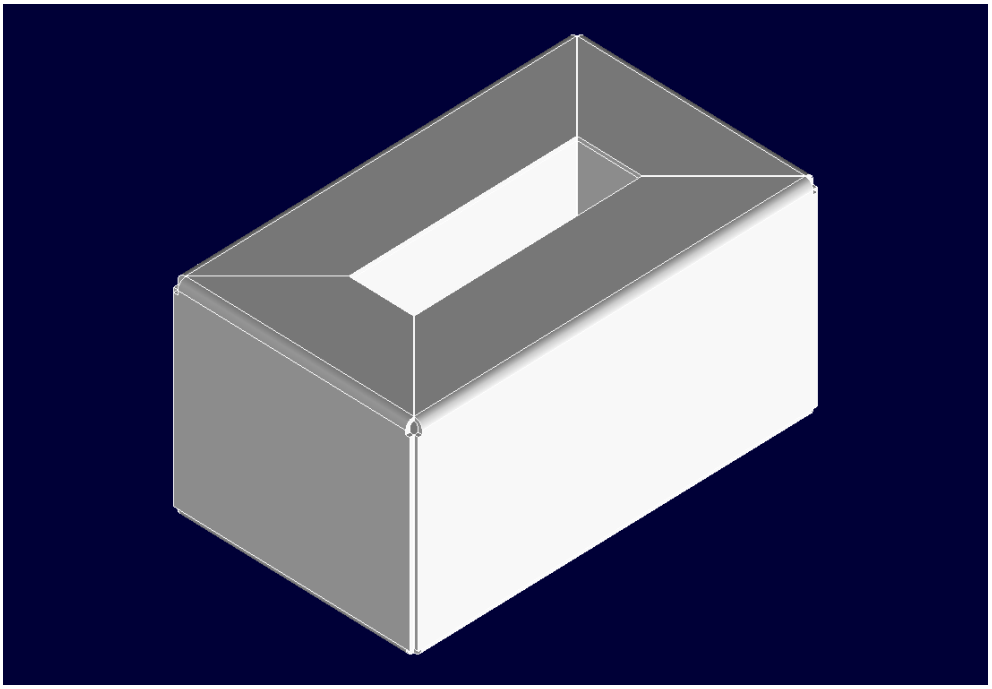


Figure 45 Four inward top flanges with 45-degree ends on sample 18

Actual sample 18 result. Straight setbacks leave clearance between the R bands.

Sample 18 recipe: thickness 1mm, R1mm, length 20mm

• Start from the box before adding top flanges. Undo a full-width return first or use the supplied starting model; this option does not shorten existing bend bands.

• Select an inner straight top edge. Choose Start / end positions and Both ends 45 deg - narrow tip.

• Set Start = 1mm and End = edge length minus 1mm. Both are distances from cyan endpoint A.

• Set radius 1mm, straight length 20mm and bend angle 90 degrees. Apply.

• Repeat on the other three edges. Picking another edge resets the range, so enter the start and end again each time.

Sample 18 edge	Start	End
Short: 57.9mm	1.0mm	56.9mm
Long: 97.9mm	1.0mm	96.9mm

The 1mm setback is verified for sample 18, not a universal rule. Different radii, thicknesses and corner geometry require different extents. This example has approximately 0.07mm clearance between the diagonal planar ends. The R bands have square setbacks, not automatically mitered curved ends.

End shapes

Square retains a rectangle. Narrow tip removes L from each tip end; the range must exceed 2L. Widen tip adds L per end, useful for outward flanges; its root width remains unchanged and neighboring clearance needs separate adjustment. Part rules still control rectangular, round or no relief.

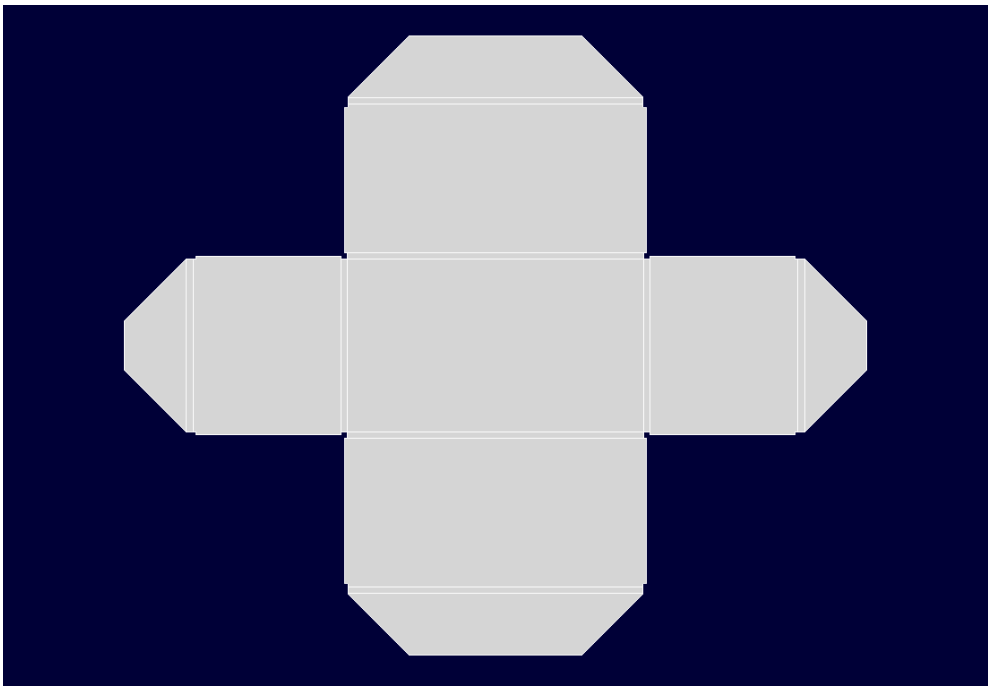


Figure 46 The four trapezoid flanges in the unfolded blank

Verified unfold/refold and save/reload with all three relief types.

Use regular Taper faces for subsequent end-face edits. This option does not automatically trim neighboring flanges or calculate their gap; interfering material is rejected.

Bend along line: parts with existing bends

Sheet-metal settings · 日本語

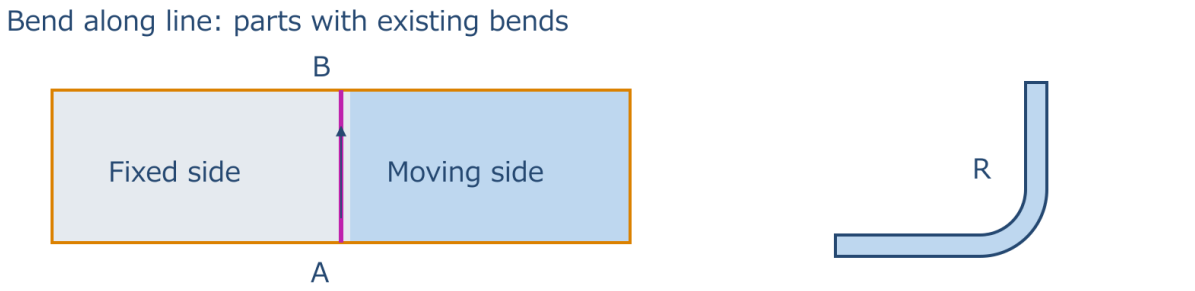


Figure 47 Green = solid outline; orange = face outline; magenta = bend line

Machined bends, automatic miters and dimension references

Add a line bend to a sheet-metal part that already contains edge flanges or line bends. Earlier flanges attached to the moving region travel with that region.

Steps

- Draw a straight line across a planar panel, from boundary to boundary.
- Open Bend along line and pick the solid. Only its green outline is highlighted, leaving surface curves visible.
- Pick the planar face containing the line under Fixed face. Its outline is orange.
- Pick the Bend line. It is highlighted in magenta.
- Set inside radius R and angle, then click Apply.

Direction

Looking straight at the selected face, the right side of the line's start A → end B moves. The left side stays fixed. The face pick location does not choose the fixed side; the line direction does. Reverse the source line to exchange sides.

A bend allowance strip is reserved on the moving side using thickness, inside radius, K factor and angle. Attached flanges travel with their panel.

Supported operations and limits

- Planar base, flange and previously bent panels are supported.
- Unfold/refold, save/reopen and Undo/Redo preserve bend relationships. Through cuts added to flat regions follow their corresponding panels when refolded.
- A new bend strip must not cross a hole, an existing curved band or bend attachment.
- The line must separate the panel; collisions after folding are rejected. Refold an unfolded part before adding a bend.

Geometry is computed only on Apply. Failed operations leave the original solid intact.

Edit bend / flange

Changing a parent retains common blank coordinates and requires a valid connected result. Cycles, detached material and overlap are rejected. This does not relocate a flange to an arbitrary new attachment.

Imported recognition and ordinary editing

Recognize → Existing cylindrical bends follows tangent plane/cylinder connections from a broad stationary face. Opposing skins must have consistent thickness. Holes and cutouts remain, and the complete reconstructed exterior must match the source. Rip closed loops first. General freeform pressings are excluded.

Bend along line · 日本語

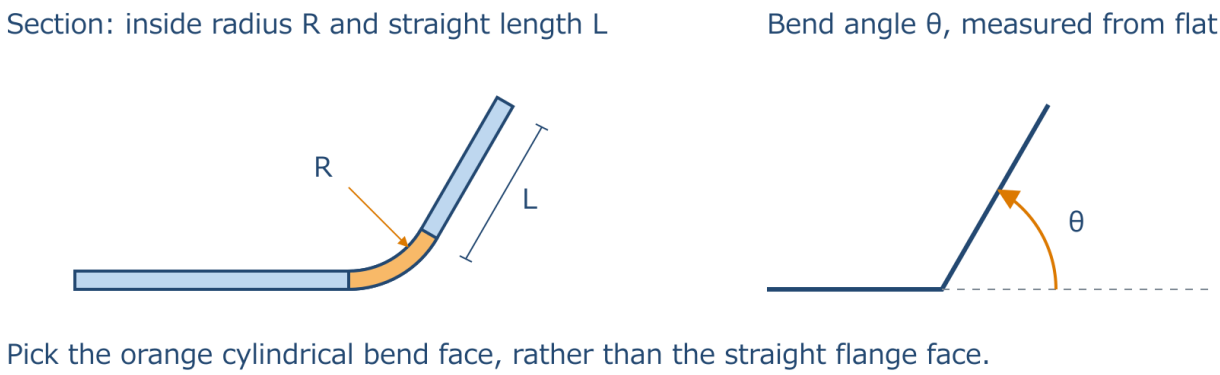


Figure 48 Pick the orange cylindrical bend face, rather than the straight flange face.

Machined bends, automatic miters and dimension references

Edit the angle and inside radius of bends created with Edge flange or Bend along line. Ordinary edge flanges also support editing straight length L.

Steps

• Open Edit bend / flange and pick the sheet-metal solid.

• Under Bend face, pick the curved cylindrical face at the root. Either the inside or outside face works. The current angle, radius and length are loaded.

• Change Bend angle, measured from the flat state. Change inside radius R if needed.

• Click Apply. Only a successful calculation replaces the solid. Undo restores the previous shape.

Length of a bend created from a line

The flat blank outline and hole positions stay fixed.Changing angle or radius changes the bend allowance, leaving a different straight length L. The L field is disabled and labelled as calculated on Apply. It shows the previous length during input and the calculated result after Apply.

Bend allowance = angle in radians × (inside R + K factor × thickness t). This edit preserves K factor, thickness and the fixed side.

Ordinary edge flanges allow direct entry of L. If a further line bend has split their planar panel, L is calculated to preserve the blank and the additional bends. Bends attached to the moving region travel with it.

Limits

• Angle must be greater than 0 and less than 180 degrees; inside radius must be positive.

• Crossing another bend attachment and collisions after folding are rejected. Holes and notches retain their developed positions.

• Refold unfolded parts first. Synchronized regular face edits and accepted planar through cuts are retained. Angle/radius editing retains holes and notches after machining a bend band. See regular sheet-metal editing.

The illustration explains parameters; it is not a live preview. Geometry is calculated only on Apply, not while selecting faces or entering values.

Existing reliefs of a bend with calculated length remain as notches in the developed outline when editing angle or radius.

Burring, dimples, embosses and beads

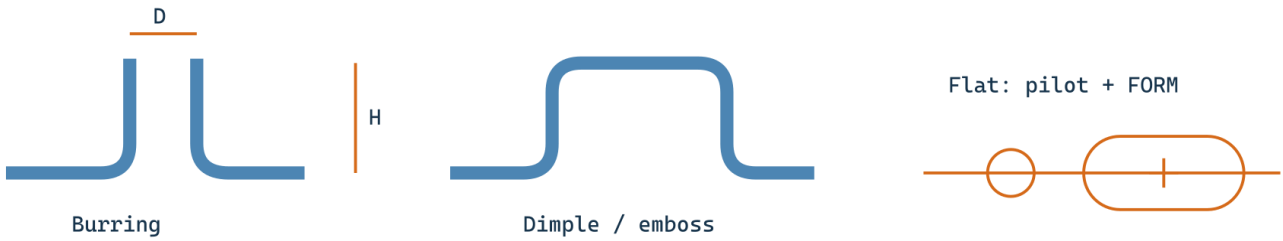


Figure 49 Form sections and flat representation

Select a solid, a planar face and a center point. Enter the type, dimensions, rotation and direction, then Apply. The full root footprint must fit on one panel, clear of holes, bends and other forms.

Type	Dimensions	Flat representation
Burring	D = bore diameter; H = height from the selected face; R = inside root radius	The explicitly entered pilot hole remains.
Dimple	D = flat crown diameter; H = height; root and crown radii	Original flat material with a forming marker.
Emboss / bead	Obround crown. D = crown width; span = distance between semicircle centers	Flat material and the forming footprint, center and note.

Profiles maintain constant geometric thickness. The crown radius must exceed thickness, and height must accommodate both radii. Burring height must accommodate its root radius. Edit, suppress, resume or delete a form by F number (creation order).

Plastic forming cannot be reversed into a unique production blank from the finished shape alone. Supply the pilot diameter for your material and tooling. SM_FORM, SM_FORM_CENTER and SM_FORM_NOTE represent forming instructions; they are separate from SM_CUT.

Ordinary through-holes away from the form and compatible outline edits are retained in either state. Editing the formed surface itself, cutting its root, or placing a form on a curved surface is outside this workflow.

Thickness and bend compensation

Use Sheet-metal settings → Selected part to change thickness while retaining the actual blank outline and holes. Refold partial developments first. Changes that create interference are rejected.

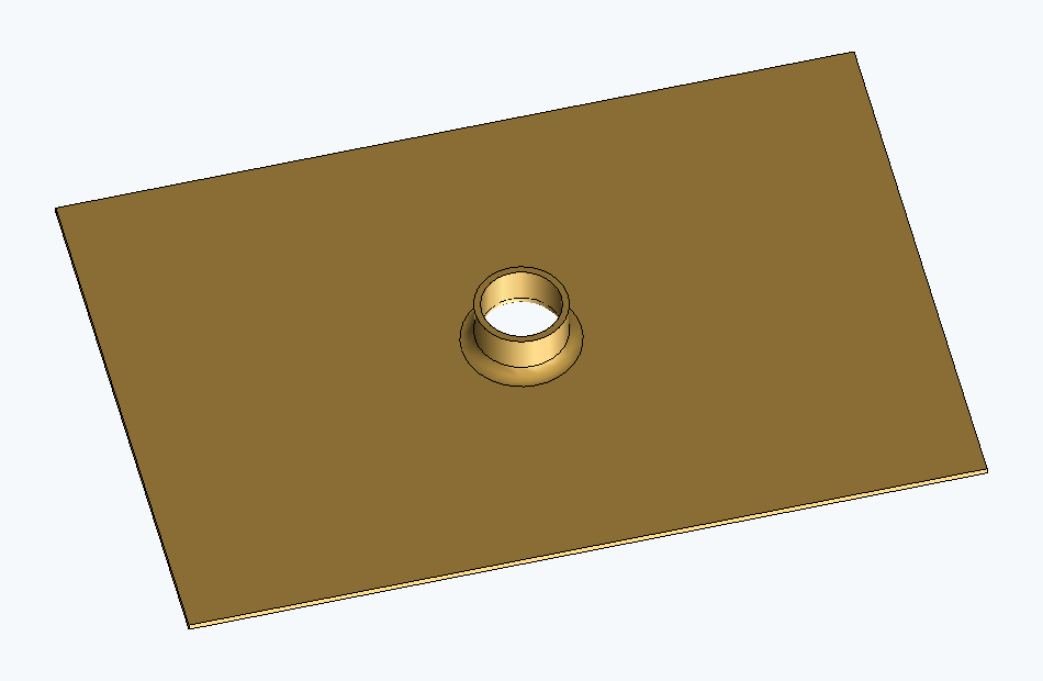


Figure 50 Native burring result

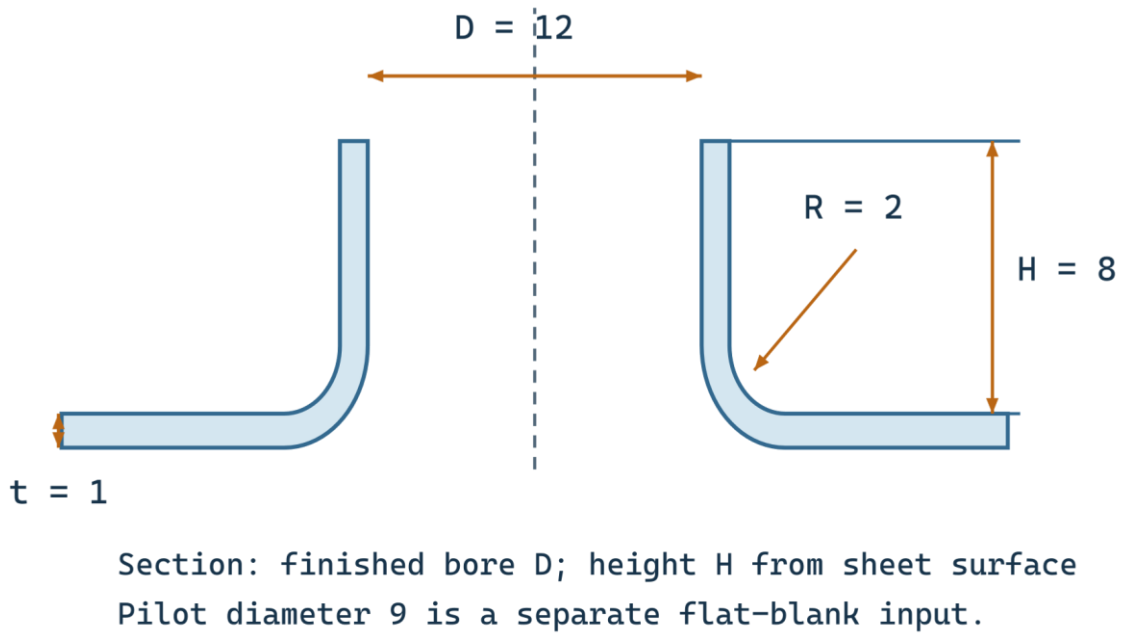


Figure 51 Section showing D, H, R and t

1. Create one burring

• Prepare a recognized sheet-metal base plate. Example A is 120 × 80 mm with thickness 1 mm. Recognize an existing solid first if needed.

• Open Sheet metal → Forming. Choose Create and Burring.

• Use Select in CAD for Solid, then select the complete plate.

• Use Select in CAD for Base face, then select the broad top face, not the thin side face.

• Pick the forming center on that face. The example is 60 mm in X and 40 mm in Y from the plate’s lower-left corner.

• Enter bore diameter D=12, height H=8, root radius R=2 and pilot diameter=9.

• Choose Positive direction. Leave rotation at 0° for this circular burring example.

• Apply to calculate and validate. A successful result is one solid and one Undo step. Close the dialog after checking the result.

Start from an uncut planar region. Do not pre-drill the pilot hole: an existing hole within the root footprint prevents placement in the current implementation. Geometry is calculated on Apply, not while typing.

2. Dimensions and direction

Input	Meaning
D	Finished bore diameter for burring; flat crown diameter for a dimple; flat crown width for emboss/bead.
H	With Positive direction, height from the selected sheet surface to the tip, not from the sheet underside.
Root radius	Root transition. Burring requires H ≥ root radius.
Crown radius	Dimple/emboss/bead only. Crown radius must exceed sheet thickness; H ≥ root radius + crown radius.
Pilot diameter	Hole remaining in the flattened burring. Separate from finished D; 9 mm is an illustration, not a production recommendation.
Straight span	Distance between semicircle centers. Crown length = D + span. Used for emboss/bead only.
Rotation and direction	Rotation orients obround forms. Positive forms outward from the selected face; Negative forms outward from the opposite skin.

3. Six examples

All plates are 120 × 80 mm, thickness 1 mm. B/D/E/F use Positive and 0° rotation. C was formed at its final location with B's dimensions, then unfolded.

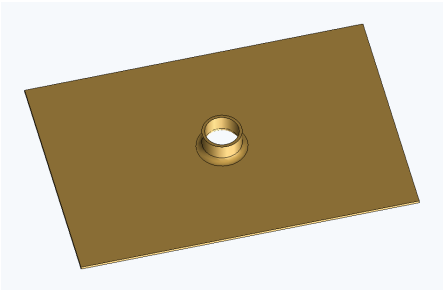
A · Plain base plate



120 × 80; thickness t=1

Use this plate to practice.

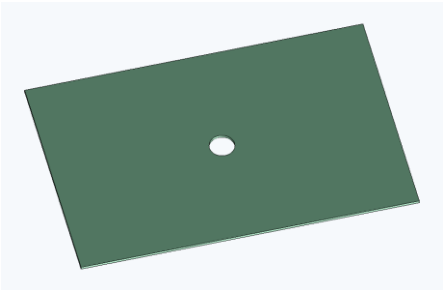
B · Burring



D12 / H8 / root R2 / pilot 9

An open collar surrounding a bore.

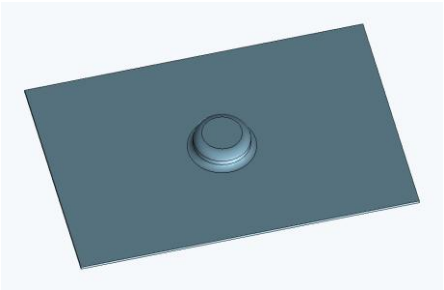
C · Flattened burring



Same dimensions as B; pilot 9

The collar is flattened; the specified pilot hole remains.

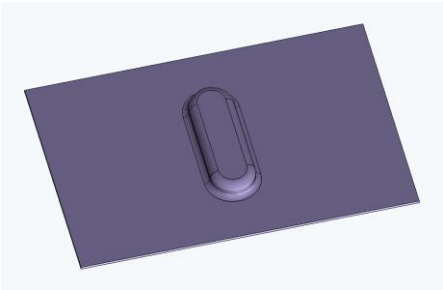
D · Dimple



D14 / H7 / root R2 / crown R3

A round form with a closed crown.

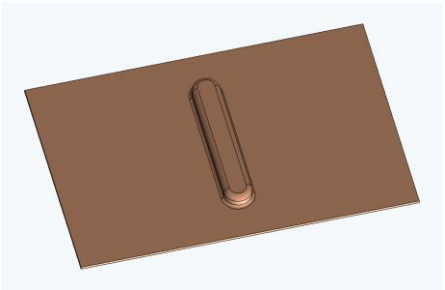
E · Emboss



D10 / H6 / root R2 / crown R3 / span 30

An obround crown; crown length is D + span.

F · Bead



D6 / H5 / root R1 / crown R2 / span 45

A narrow example; currently uses the same obround construction as emboss.

4. Unfolding, editing and troubleshooting

Unfold and refold	**Unfold replaces burring with its pilot hole; the other forms become uncut flat material. Refold restores the formed shape.**
Flat-pattern drawing	SM_CUT contains cutting boundaries. SM_FORM, SM_FORM_CENTER and SM_FORM_NOTE describe forming; do not treat these as cutting paths.
Edit dimensions	Choose Edit dimensions, select the solid and F number, adjust values and Apply. Each example has one form, F1. Placement/direction are create-only inputs.
Suppress or delete	Suppress retains the feature definition and restores flat material; Resume restores the form. Delete removes its definition as well.
Placement failure	The complete root must fit on one planar panel, clear of edges, holes, bends and other forms. New forms on curved/rolled or unfolded parts are unsupported.
Invalid dimensions	Use positive dimensions. Burring: H ≥ root radius. Other forms: crown radius > thickness and H ≥ root + crown radii.
Model scope	Constant-thickness finished geometry, not material-stretching, thinning or fracture simulation. Choose production pilot diameters using actual material/tooling conditions.

Louvers, drawn forms and bridges

Select the type in Sheet metal > Forming. Requires STD/PRO. Each result is one constant-thickness solid joined to the plate. Geometry is calculated on Apply, not while typing dimensions.

Flat restoration does not calculate a production draw blank. Louver/bridge slits are drawing paths; cutting kerf and plastic stretch allowances are not calculated.

Common workflow

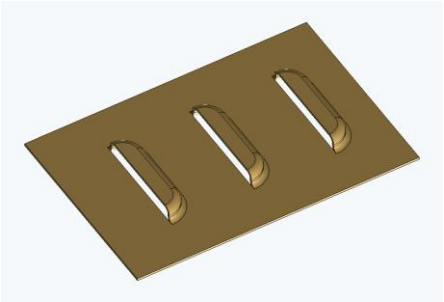
- Create a 120 × 80 mm base plate, thickness 1 mm. Recognize an existing solid as sheet metal first.
- Open Sheet metal > Forming and select Create.
- Pick the solid, a broad planar reference face, then the form center.
- Set type, dimensions, rotation and direction, then Apply. Do not drill a pilot beforehand.
- Unfold restores the reference flat sheet. Refold restores the form. Use Create/update flat pattern to see shear slits.

1. Louver: an open-sided vent

The placement point is the slit midpoint. D is flat roof depth; L is the straight opening span; H is top height from the surface. Clear opening height is H minus thickness. Rotation sets the opening direction; zero opens toward the selected face's local -X, not necessarily screen left.

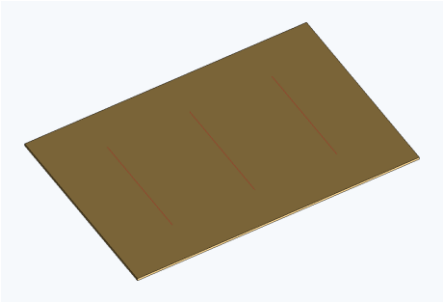
G: three centers at -35, 0 and 35 mm from plate center. H: flat. I: refold. Slit length = L + 2 × (D + root R + shoulder R), 46 mm in this example.

G · Louver / Formed



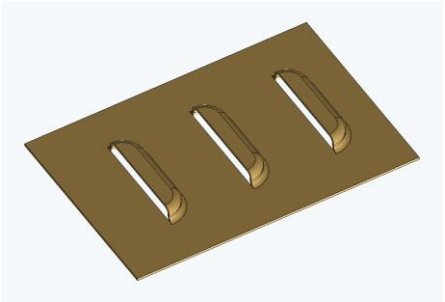
D6 / H6 / root R2 / shoulder R3 / L24

H · Louver / Flat



D6 / H6 / root R2 / shoulder R3 / L24

I · Louver / Refolded



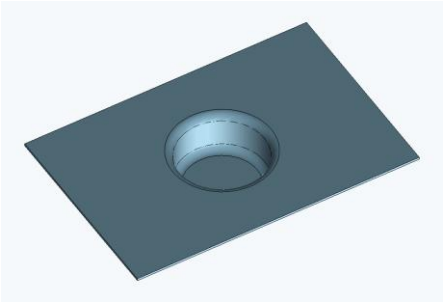
D6 / H6 / root R2 / shoulder R3 / L24

2. Draw: round cup and rounded rectangular tray

Round draw shares the existing dimple's constant-thickness section. Reverse makes a recessed cup. Rectangular draw has a rounded rectangular bottom and four walls; its plan corner R is separate from root/shoulder R. D/W/L describe the flat bottom, not the outside opening.

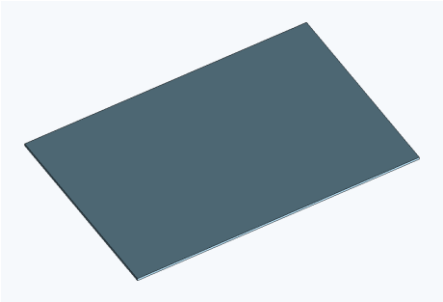
J/K/L: round cup, flat, refold. M/N/O: rounded tray, flat, refold. Both use reverse direction.

J · Round draw / Formed



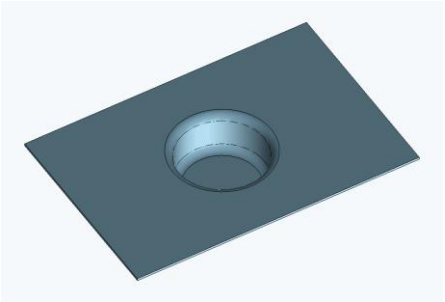
D30 / H18 / root R2 / shoulder R4 / reverse

K · Round draw / Flat



D30 / H18 / root R2 / shoulder R4 / reverse

L · Round draw / Refolded

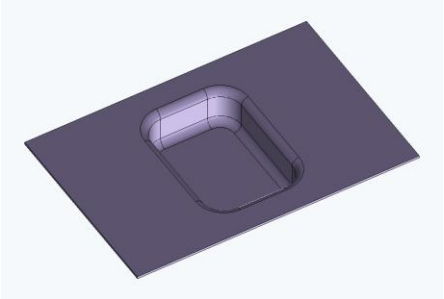


D30 / H18 / root R2 / shoulder R4 / reverse

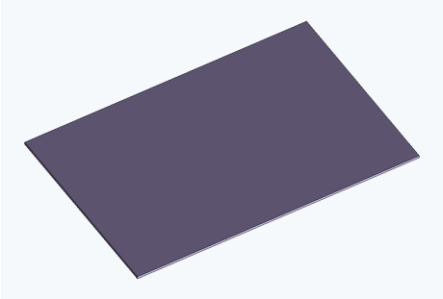
M · Rectangular draw / Formed

N · Rectangular draw / Flat

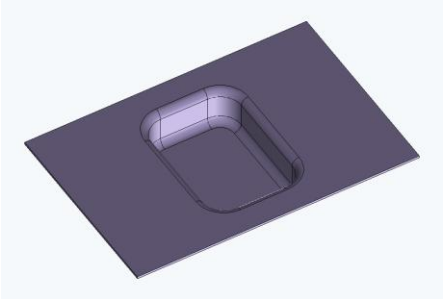
O · Rectangular draw / Refolded



W32 / L48 / H16 / plan R6 / root R2 / shoulder R4 / reverse



W32 / L48 / H16 / plan R6 / root R2 / shoulder R4 / reverse



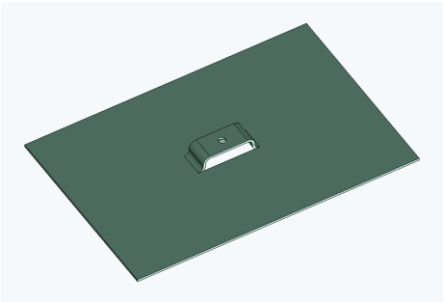
W32 / L48 / H16 / plan R6 / root R2 / shoulder R4 / reverse

3. Bridge: two attached ends, two sheared sides

Raise a strap while retaining both ends. D is flat roof length, L strap width, H top height. A center hole diameter of zero disables the hole. A positive hole must be smaller than both D and L.

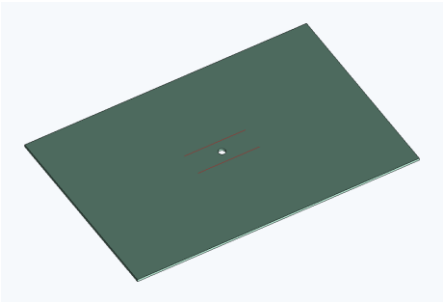
P/Q/R: single bridge, flat, refold. S/T/U: double bridge, flat, refold, with centers 46 mm apart. Create two separate forms; there is no special double-bridge type.

P · Single bridge / Formed



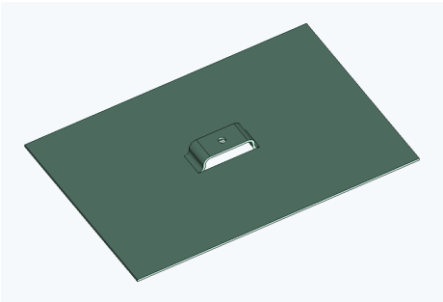
D16 / L10 / H6 / root R2 / shoulder R3 / hole 3

Q · Single bridge / Flat



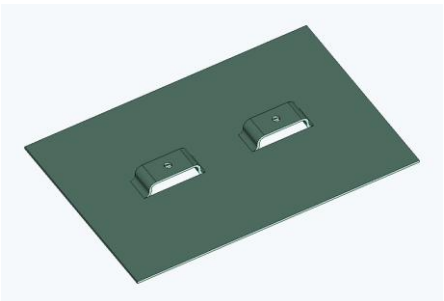
D16 / L10 / H6 / root R2 / shoulder R3 / hole 3

R · Single bridge / Refolded



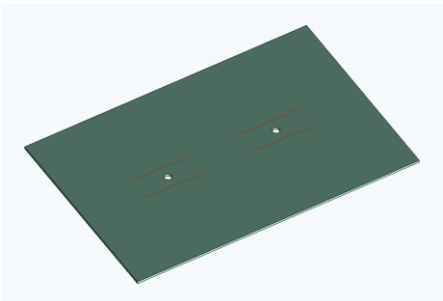
D16 / L10 / H6 / root R2 / shoulder R3 / hole 3

S · Double bridge / Formed



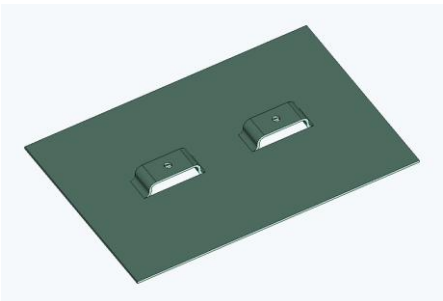
Same dimensions / center spacing 46

T · Double bridge / Flat



Same dimensions / center spacing 46

U · Double bridge / Refolded



Same dimensions / center spacing 46

Dimensions, unfolding and editing

Radii and height	**Root R > 0; shoulder R > thickness; H >= root R + shoulder R. Rectangular width/length must exceed twice plan corner R.**
Placement	Keep the full placement footprint within one face, clear of existing holes, edges, bends and other forms. A louver also needs flat material in front of its opening.
Required footprint	Louver: obround with radius D + root R + shoulder R + thickness and straight span L. Bridge: rectangle D + 2 × (root R + shoulder R + thickness) by L + 2 × thickness.

Radii and height	**Root R > 0; shoulder R > thickness; H >= root R + shoulder R. Rectangular width/length must exceed twice plan corner R.**
Flat shear paths	SM_CUT contains one zero-kerf slit per louver or two per bridge. The 3D flat sheet has no visible-width slit groove. Bridge center holes remain in the flat solid.
Layers	SM_CUT contains cut/slit paths. SM_FORM, SM_FORM_CENTER and SM_FORM_NOTE are forming outlines, centers and notes; do not cut those outlines.
Drawing limitations	Unfold restores the reference blank. It does not calculate material flow, thinning, cracking, process stages or a production draw blank, and does not guarantee tooling feasibility.
Edit and supported parts	Edit dimensions, suppress, resume or delete using F numbers. Create on planar regions of base/folded sheet metal; not on unfolded parts or curved roll sheets.
Already running CAD	Samples can be displayed, but operating new form types and unfolding/refolding their history requires the updated build at the next start. The active CAD process is not replaced automatically.

DLL / MCP

sheet.form / sheet.editForm: kind=4 Louver, 5 RoundDraw, 6 RectangularDraw, 7 Bridge. diameter = D/W, span = L, cornerRadius = rectangular bottom plan R, bridgeHoleDiameter = center hole diameter (0 = none).

Bend relief: create, edit and remove

Create relief at the end of an existing sheet-metal bend. This command also edits the width, depth and shape of registered reliefs, and removes them. It does not cut arbitrary straight edges of an unbent plate.

Purpose	Action	Result
Create	Select an unrelieved bend end, choose Rectangle or Round and enter dimensions	Add relief at that end
Edit	Reselect the same bend end and apply new dimensions or shape	Replace its registered relief
Remove	Select the bend end and Apply with None	Remove its registered relief

1. What to select

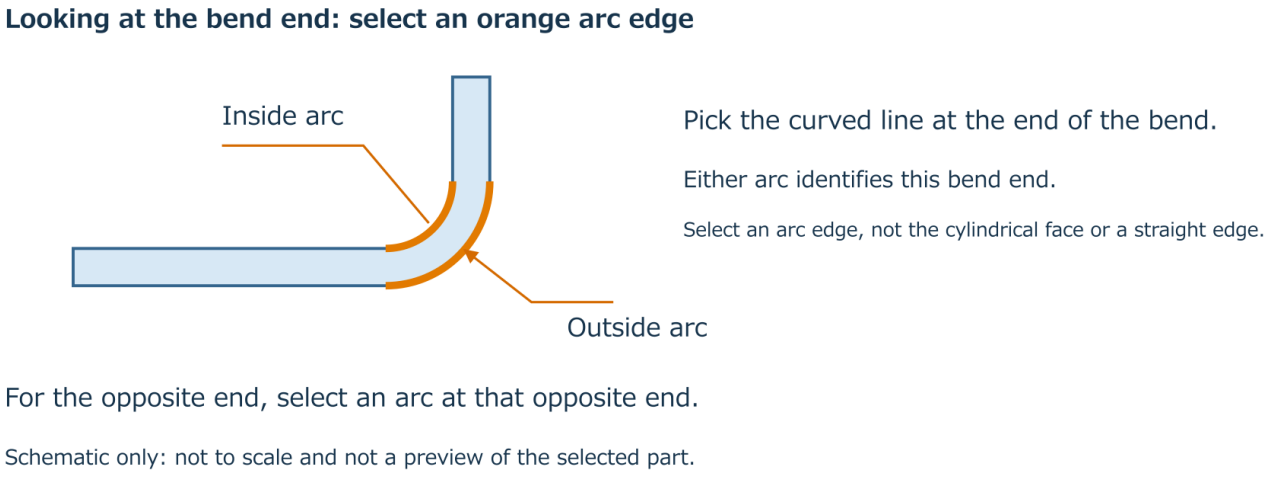


Figure 73 Inside and outside bend-end arcs highlighted in orange. Select either arc edge.

The arc is where the cylindrical bend meets its end section. After relief has shortened a bend, select its currently visible arc. This is not the rounded edge at the back of the relief notch.

2. Create a relief

- Open Bend relief and select the folded sheet-metal solid. Refold it first if unfolded.
- Under Bend end edges, choose Select in CAD and pick the end arc. Selected edges are orange.
- Choose Rectangle or Round, then enter Relief width w and Relief depth d.
- Apply. Selection and value entry alone do not modify the solid.

Multiple selected ends receive the same settings. Clicking an already selected edge deselects it. Applying clears the edge selection; select the new arc again for another edit.

3. Edit an existing relief

- Select the solid and the current arc at the desired bend end.
- Enter the new width, depth and shape. Rectangle and Round can be exchanged.
- Apply. The registered relief at that end is replaced; there is no need to remove it first.

Existing relief dimensions are not automatically loaded when an edge is selected. Enter the desired new values; do not treat the displayed defaults as a measurement. Edit ends separately if they need different settings.

4. Dimensions and shapes

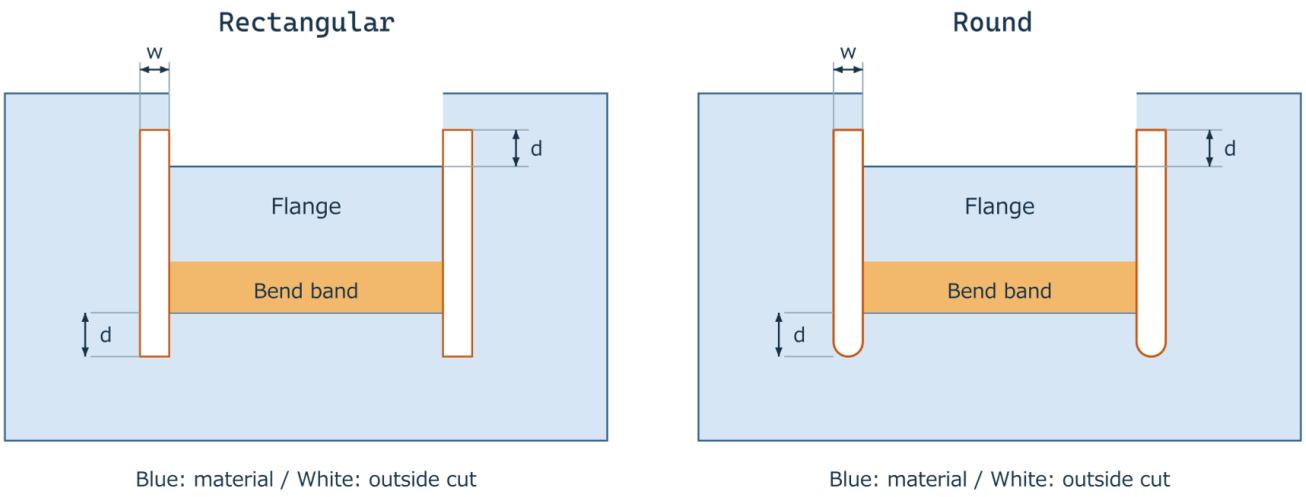


Figure 74 Developed concave flange with width w and depth d cut in the neighboring sheet.

The dialog illustrates a concave flange relief. White is removed material; this is not a live preview.

Input	Meaning
Width w	Notch width. A concave flange cuts the neighboring sheet outside the bend/flange span, retaining that span.
Depth d	Depth into the planar material from the bend boundary. For concave outside relief, this is depth into the base, with the mouth extended beyond the developed flange tip.
Rectangle / Round	A squared notch or one with a rounded inner end. The mouth remains open to the outside.
None	Remove registered relief at that end. Width and depth are unused; do not set dimensions to zero to remove a relief.

Free ends and line bends use a different arrangement, cutting into the bend end and both neighboring planar regions. See the settings shape diagrams.

5. Which existing features can be edited?

• Reliefs registered by this command or automatically during flange creation are supported.

• Box-conversion seams and root openings, and notches created by ordinary cuts or face edits, are not automatically recognized as w/d relief features. Use regular face editing for separate wall/bend-end adjustment.

• After ordinary editing incorporates a relief into general material geometry, this command cannot recover its previous relief dimensions or remove it as a registered relief. Machining through bend bands can also prevent these sheet-specific parameter edits.

6. If changing values does not change the part

Check the status message after Apply. Invalid settings abort the operation and retain the original geometry. Changing a value or shape alone does not modify the part.

Selection: Pick the short arc at either bend end. The long straight edge along the bend cannot be selected. Round: d > w/2 is required, so check depth after increasing width. This particular condition does not apply to Rectangle.

Use positive w and d. The current implementation requires w smaller than half the bend span and d smaller than the straight flange length. Round requires d > w/2. Disconnected or interfering results are rejected even within these limits. Failed operations retain the source solid.

Removal also validates the restored material. None does not fill arbitrary holes or ordinary cutouts. Successful changes support Undo/Redo.

Create rip: box corners and planar slits

Sheet-metal settings · 日本語

Create corner openings and finite through slits in ordinary/imported solids and JDesignCAD sheet-metal parts, directly in model space.

Choose a method

Method	References and use
Box corner (edge)	Solid + one inside or outside vertical corner edge between perpendicular planar walls.
Box corner (two faces)	Solid + two broad wall faces. Works with separate flanges without a shared edge. Pick order defines sides 1 and 2.
Slit from a line on a face	Solid + planar face + a straight line drawn on it. Interior slot or open notch, also on planar walls of bent parts.
Outer-edge trim (legacy)	Flat/unfolded sheet-metal part + straight outer edge. Removes a strip; on a simple plate this resembles moving its end face inward.

Open a box corner

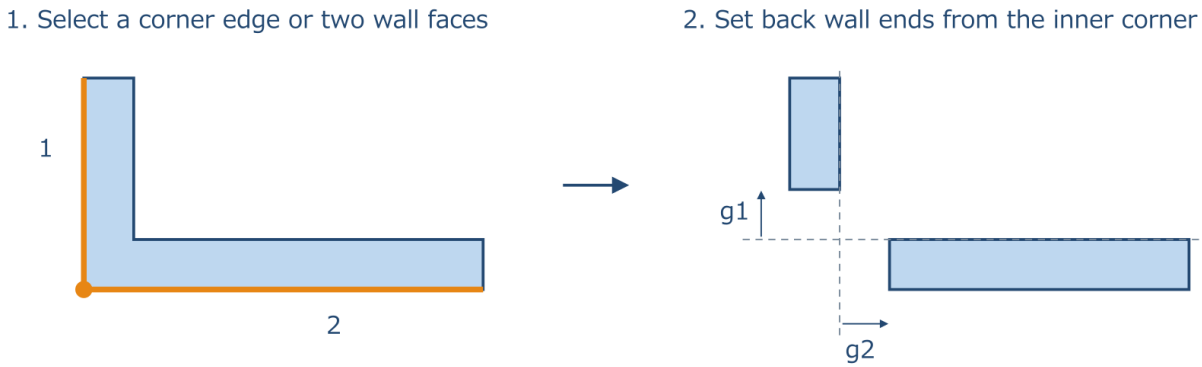


Figure 75 Top section. Dashed: inner corner. g1 / g2: setbacks. Symmetric: g1 = g2 = g/2.

• Choose a corner mode and select the solid.

• Pick one vertical corner edge, or two broad wall faces in sequence. Pick two inner faces or two outer faces, not narrow thickness faces.

• Set gap g and its side. Start with Symmetric.

• Apply. The floor and curved bend bands are retained; the planar wall region is opened.

g allocates wall-end setback from the inner corner: Side 1 gives g1=g, g2=0; Side 2 gives g1=0, g2=g; Symmetric gives g1=g2=g/2. It is not the shortest diagonal distance. No material is added to an already open corner; a smaller setting may produce no change.

Edge mode uses internal face order. Use two-face mode to control side order. Rounded corners, non-perpendicular walls and cuts across curved bend bands are not supported.

Make a slit

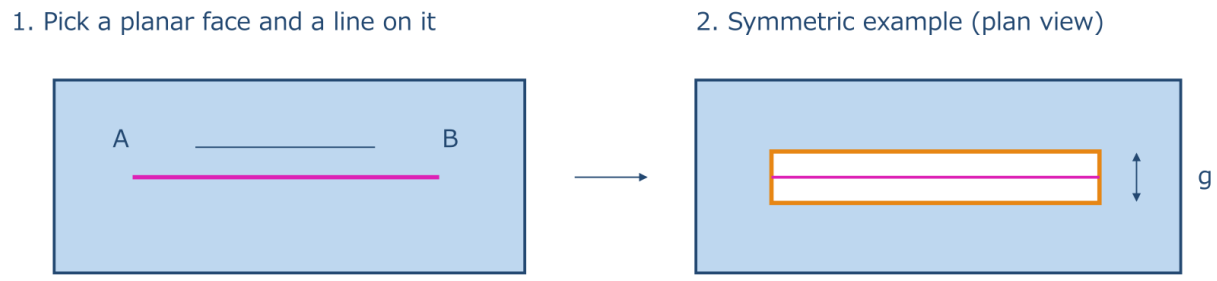


Figure 76 Magenta: line A to B. White: through slit. g: gap width. The original line is retained.

• Draw a straight line from A to B on the desired face.

• Choose the slit mode.

• Pick Solid, Slit face, then Slit line.

• Set g. Viewed from outside the selected face along A to B, Side 1 is right and Side 2 is left; Symmetric uses g/2 on each side.

• Apply. The cut passes through the thickness; the line remains.

The line is not extended and ends are square. The full slot width must remain in material on the same planar face. Cuts across holes or bend bands, curved slots, round ends, and cuts separating the body into multiple solids are unsupported.

Sheet-metal history and imported solids

For a part with JDesignCAD sheet-metal history, planar removals are stored in flat coordinates and retained through Unfold / Refold. Ordinary and imported solids can receive cuts, but cutting alone does not recognize their bends or make arbitrary bent solids unfoldable. Automatic recognition of imported bent parts is a separate capability.

Direct edits to a folded body cannot automatically be reconciled with existing sheet-metal history. Consistency is checked at Apply. Perform ordinary machining while unfolded, then Refold.

Legacy trim

Side 1 trims g inward; Side 2 points outward and removes practically no material; Symmetric removes g/2 inward. Pick a straight outer edge on the automatically recognized base face of a flat/unfolded sheet-metal part. Try the opposite broad face's matching edge if rejected. Refold does not support cuts across a bend band.

Selection and Undo

Body outlines are green, face/edge references orange, and the slit line magenta. Faces are not filled, keeping drawn lines visible. Calculation occurs only on Apply. Undo and Redo are supported. Diagrams are schematic.

Official CAD comparison — October 9, 2026

CAD	Documented features
Autodesk Fusion	Along-edge, between-two-points-on-face and entire-face rips.
Autodesk Inventor	Single-point, point-to-point and face-extent methods; cuts normal to sheet faces with gap-side control.
Onshape	Two-edge/face joints, bend or rip, edge and directional butt styles, model minimum gap.

This update supports perpendicular planar corner openings and drawn-line slits. Entire-face removal, direct two-point picking, associative joints, extensions for butt joints and arbitrary curved-surface cuts are not included.

Convert box, L and U shapes into unfoldable sheet metal

Open Convert to sheet metal and select Convert box / L / U-shaped solid. This rebuilds a sharp-cornered ordinary solid with bends at the existing walls and seams at adjoining corners, keeping its position, outside dimensions and physical thickness.

Supported input: a rectangular bottom with one to four perpendicular walls of equal height and uniform thickness. Three walls, two opposite or adjacent walls, and a single wall are supported. Full-height corner seams are retained or widened; holes, steps, tapered walls and existing bend radii remain unsupported. Native and imported solids use the same geometry checks.

Workflow

• Open the original solid and start Convert to sheet metal.

• Set Conversion mode to Convert box / L / U-shaped solid.

• Select the whole solid. Its outline is highlighted in green.

• Select the broad inside or outside bottom face. The fixed bottom is highlighted in orange. Do not select a side wall or the narrow top rim.

• Normally use Detect for thickness. Set inside radius R, K factor and corner clearance g.

• Click Apply. The command verifies the box and thickness, then bends only the existing walls by 90 degrees and opens adjoining wall corners.

• Close, save, and use Unfold.

There is no need to create rips first. Conversion opens adjoining corners automatically. Existing corner seams such as 板金 11 are supported when they run from the inside floor to the wall top. They will not be narrowed; larger requested clearance widens them. Interior slots and cuts through the bottom remain unsupported.

Result

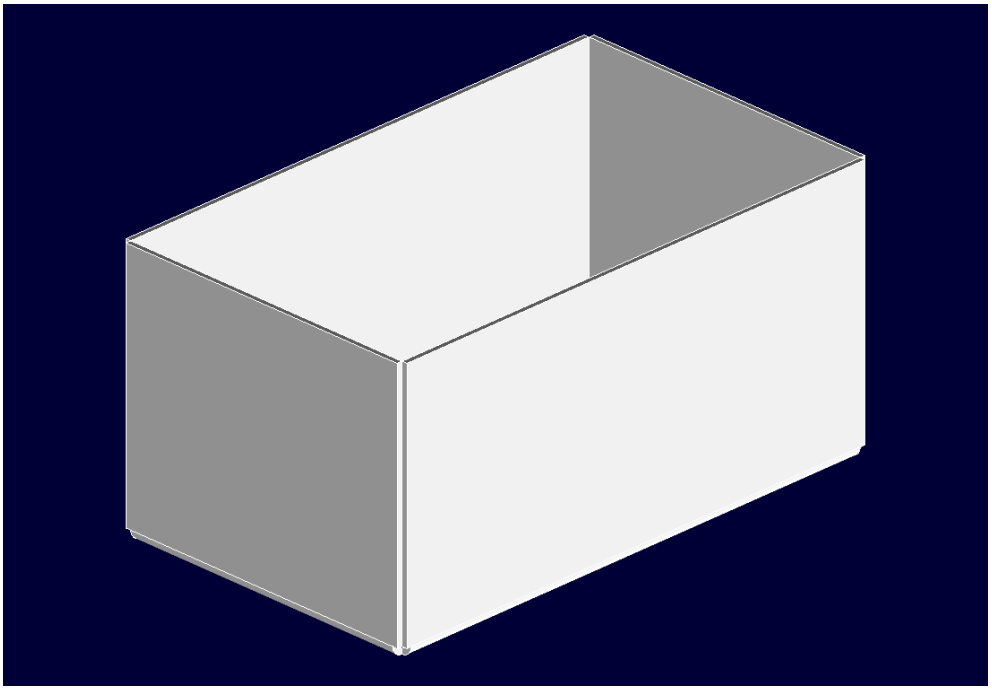


Figure 77 Converted box with vertical corner seams and bottom bends.

Wall outside planes and top edges stay fixed. The inside radius is R and the outside radius is R+t. Bend roots have automatic openings to avoid overlapping material. The original sharp corners change; all bends use the same R and K factor. Free wall ends keep their full width; no corner gap is applied there.

Wall patterns and their flat shapes

板金 12：Three walls → three bends

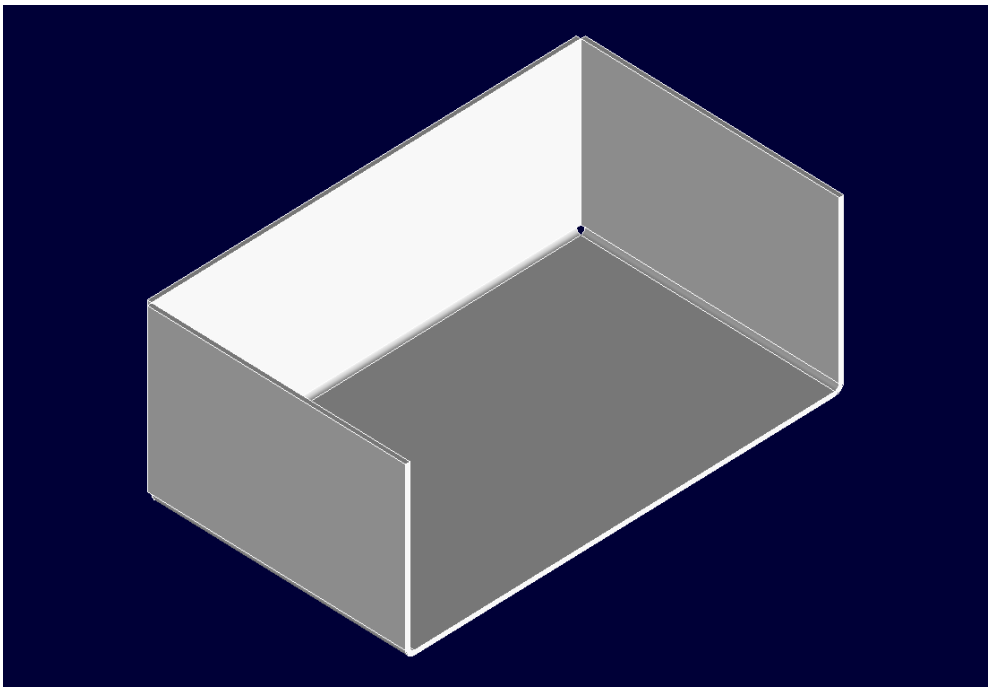


Figure 78 Converted sheet metal 12

Converted



Figure 79 Unfolded sheet metal 12

Unfolded

板金 13：Two opposite walls → two bends

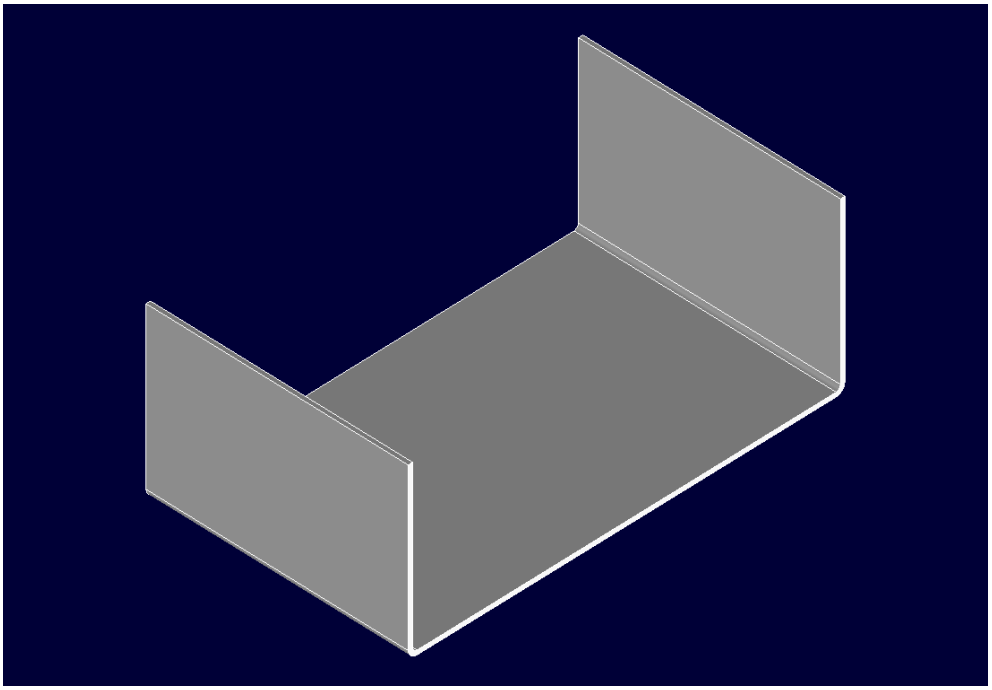


Figure 80 Converted sheet metal 13

Converted



Figure 81 Unfolded sheet metal 13

Unfolded

板金 15：Two adjacent walls → two bends

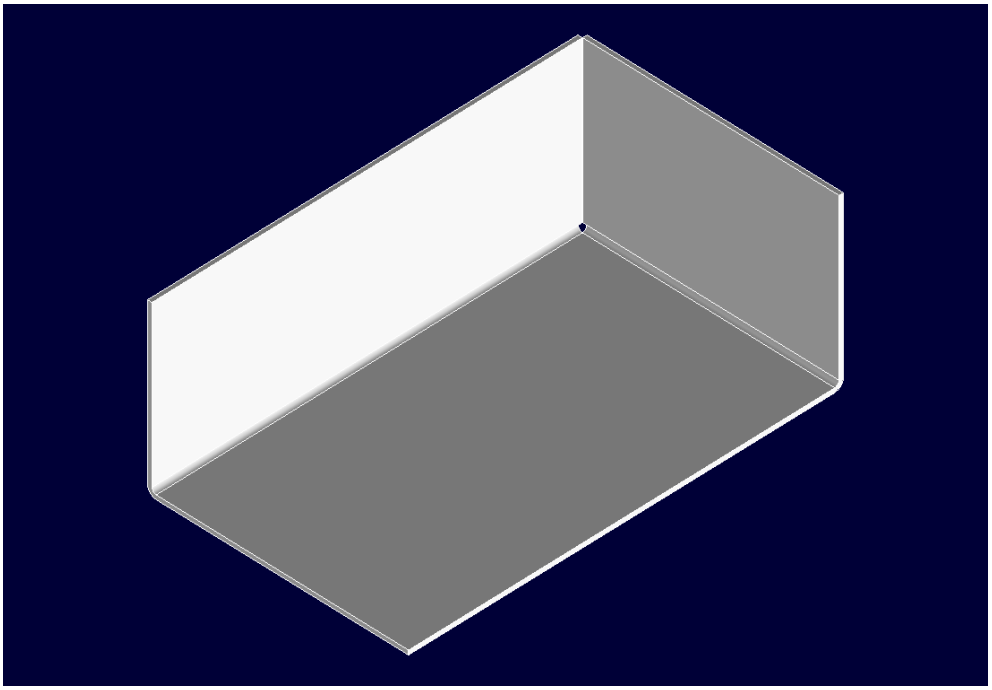


Figure 82 Converted sheet metal 15

Converted



Figure 83 Unfolded sheet metal 15

Unfolded

板金 16：One wall / L shape → one bend

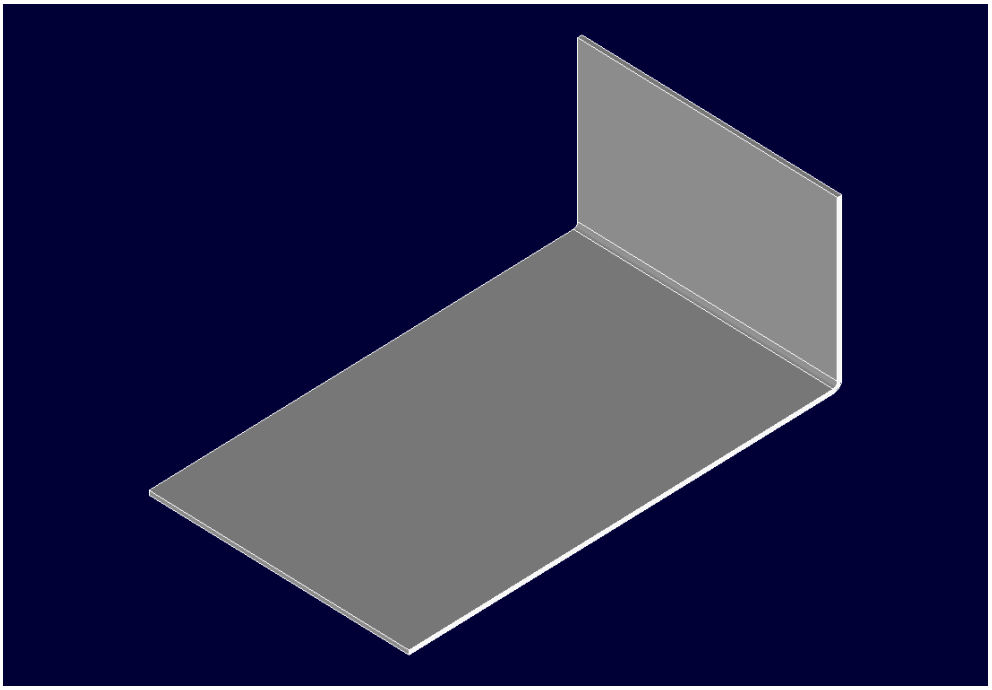


Figure 84 Converted sheet metal 16

Converted



Figure 85 Unfolded sheet metal 16

Unfolded

Parameters

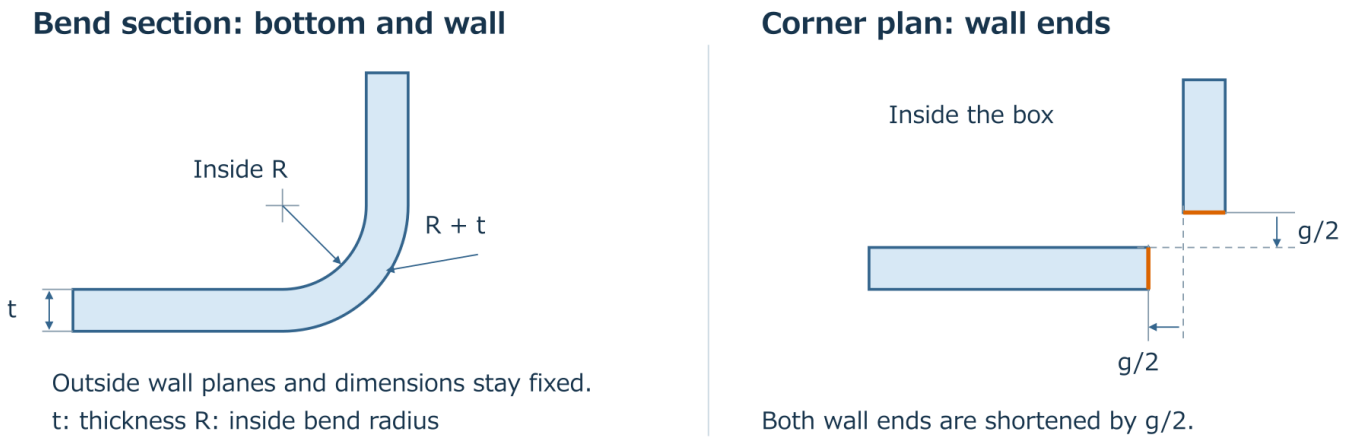


Figure 86 Bend section showing R and t; corner plan showing each wall shortened by g/2 from the adjacent inside wall plane.

Parameter	Meaning
Thickness t	The bottom and all existing walls must have the same thickness. Detect measures on Apply; the disabled value before Apply is not a measurement. Specify verifies a matching physical thickness, without resizing the box.
Inside radius R	The inner bottom-to-wall bend radius. Each bent side sets the planar bottom edge inward by R+t. Positive bottom dimensions and wall height must remain.
K factor	Determines bend allowance: each 90-degree bend unfolds to $(\pi/2)\times(R+K\times t)$.
Corner clearance g	At adjoining walls, each end stops g/2 short of its neighbour's inside plane, retaining any larger existing seam. This is not the shortest diagonal distance between the walls. Use $0<g\leq 2R$ where walls meet. For a single wall or two opposite walls, g does not affect the shape. Root openings depend on R, t and g.

Corner clearance g is separate from the width w and depth d of Bend relief. This mode uses common clearance conditions at adjoining corners and does not use the rectangular/round relief setting. To change conversion conditions, start again from the original box.

Unfold, machine and refold

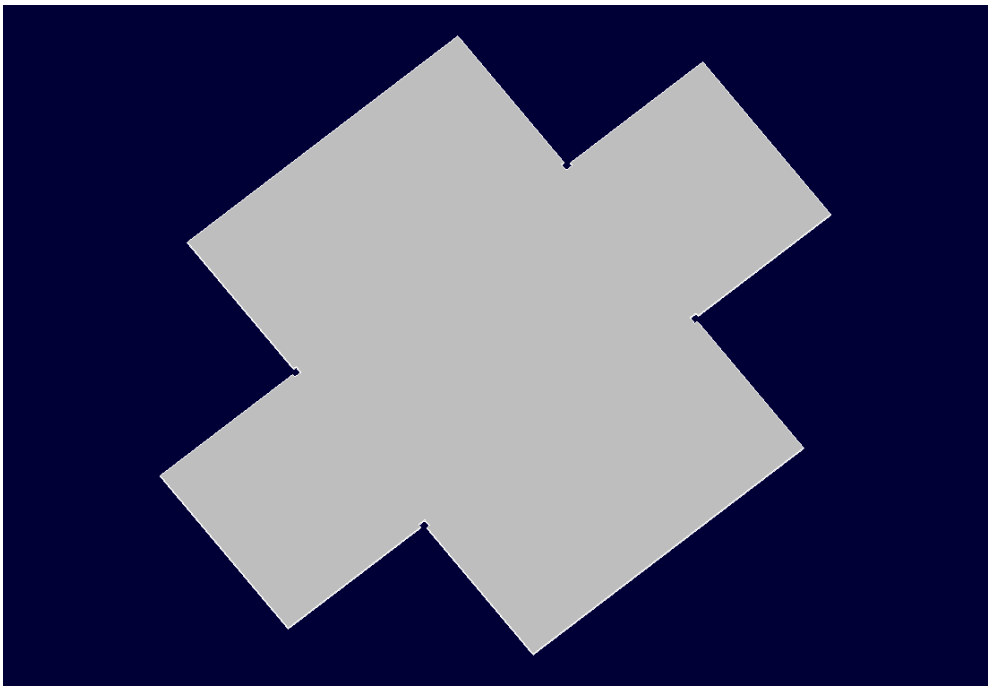


Figure 87 One connected cruciform plate unfolded from the box.

Unfold opens all registered walls in the same model space. Ordinary commands can make through holes and slots across bend bands, as well as outline edits; Refold carries them back to the folded part. The unfolded solid normally has no visible bend boundary lines. Use Create/update flat pattern for a drawing with bend lines.

Through cuts across bend bands and connected, constant-thickness outline extensions are supported. Blind pockets, thickness changes, disconnections and collisions are rejected. Editing a bend's radius or angle after conversion can change outside dimensions.

Troubleshooting

• Select a broad bottom face rather than a wall or rim.

• Use a uniform rectangular bottom and equal-height walls without holes or existing radii. Partial walls, unequal heights and angled walls are unsupported.

• Use Unfold for an already registered part; do not convert it again.

• Failed conversion keeps the original solid. Undo restores it after a successful conversion.

Changing wall and bend widths after conversion

Use regular Move faces or Offset faces on the narrow wall end or bend-end section, moving along the width direction. Only the selected end moves; the other end and planar bottom remain fixed. Select both ends to move them equally while retaining their setback. Unfolded outline editing and constant-thickness outline taper are also supported.

Illustrated steps: width edits, slots across bends and trapezoidal flanges

Corner treatment and gap dimensions

Adjust the ends of neighboring flanges to control the corner joint. Butt and 45-degree miter joints use different face selections and different definitions of gap g.

Choose the joint and faces

Joint	Faces to select	Gap g
Butt	One broad face from each of two adjacent side walls sharing a base-plate corner. Do not select both sides of one wall.	Total end retreat: each end retreats by g/2.
45-degree miter	At least two broad, coplanar faces on neighboring perpendicular flanges. All four top rims may be selected together.	Width measured perpendicular to the diagonal joint.

Do not select thin thickness faces or cylindrical bend faces. Corner treatment is separate from filleting, bend relief, welding and boolean union.

A Butt joint

The base plate is 80 by 50 mm, with thickness 2 mm, inside radius 3 mm, a 20 mm straight wall section and a 90-degree bend. Select the two highlighted side walls and set g to 4 mm.

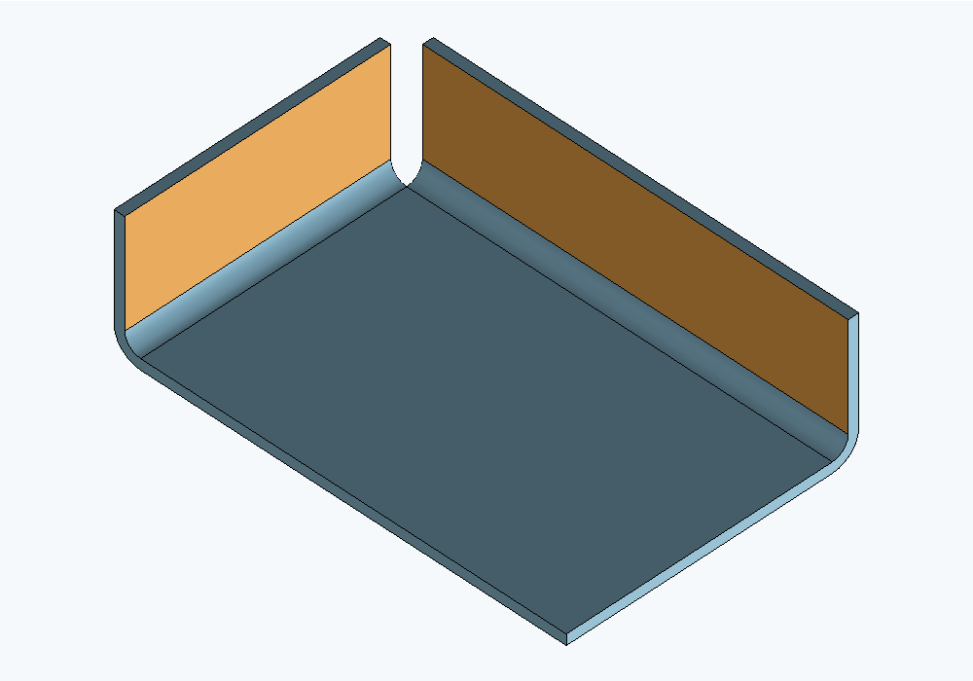


Figure 88 Before butt treatment. Select one broad face on each orange side wall.

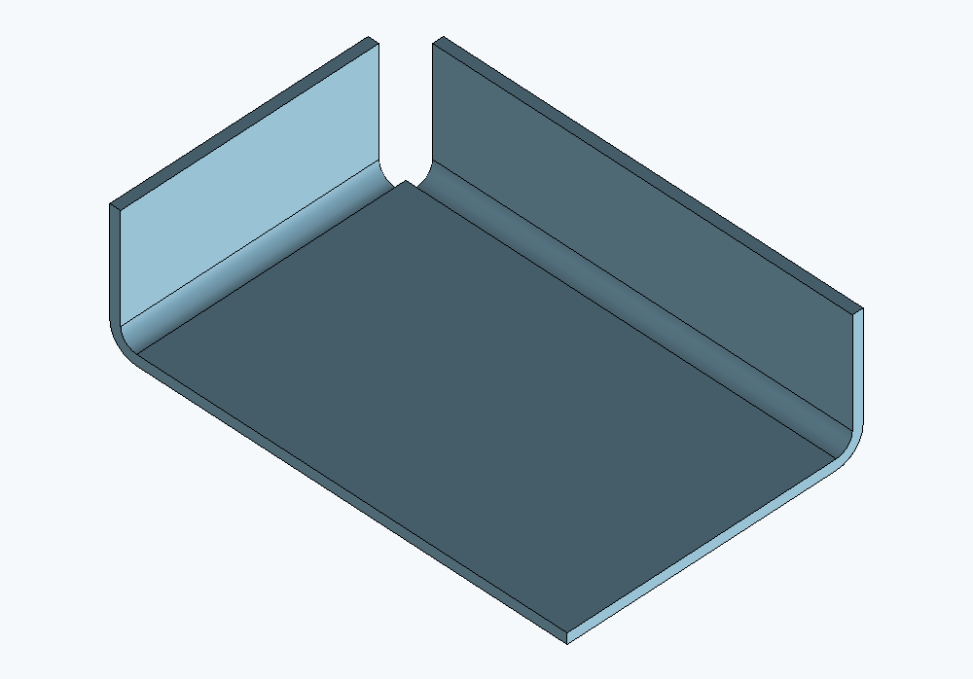


Figure 89 After butt treatment. Each corner end retreats by 2 mm.

A gap value of 4 mm does not mean that the shortest diagonal distance becomes 4 mm. Distinguish the original opening caused by radius and thickness from the added end retreat.

B Coplanar 45-degree miter

The box is 100 by 60 mm, thickness 1 mm, inside radius 1 mm, with 20 mm straight top flanges. The starting shape already has a 0.2 mm miter gap. Select all four orange top faces and change g to 2 mm.

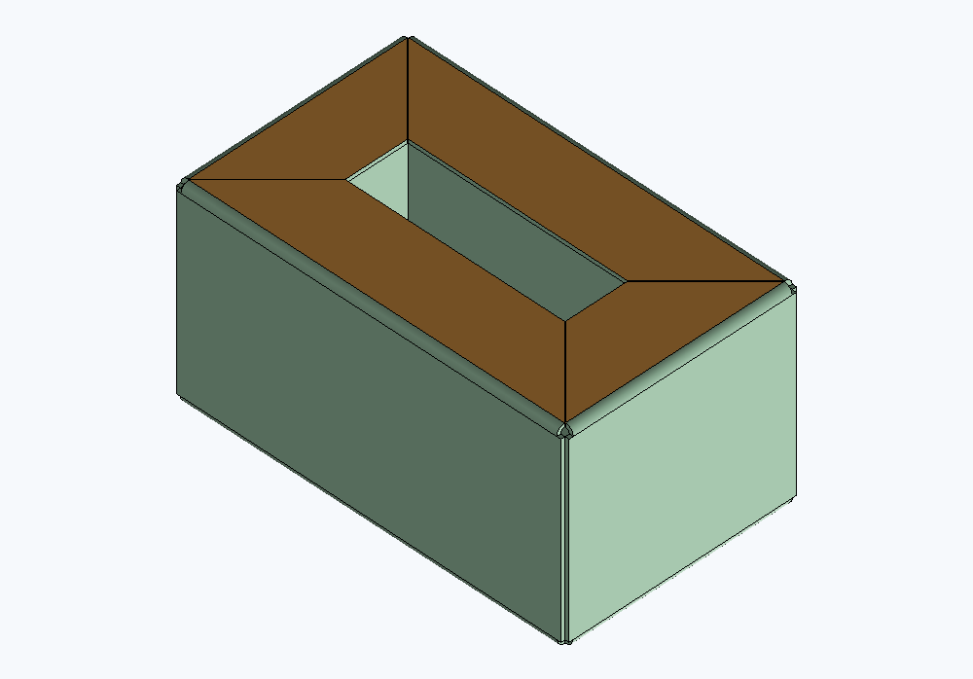


Figure 90 Before: four orange coplanar top rims with 0.2 mm diagonal gaps.

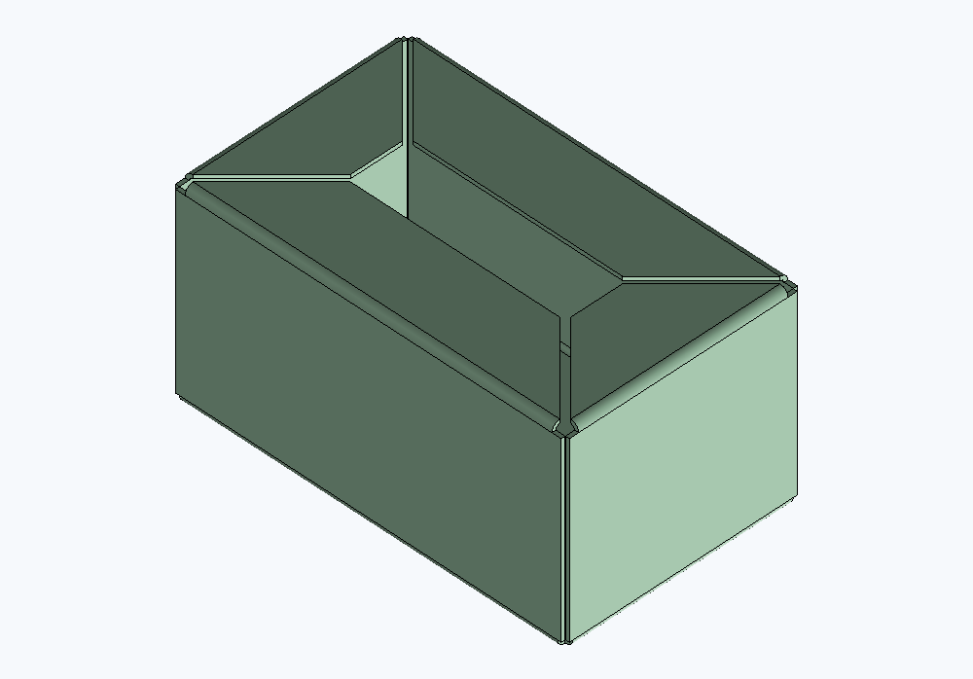


Figure 91 After: the diagonal gap is 2 mm at all four corners.

Both inward and outward flanges are supported, and the gap can increase or decrease. Arbitrary joint angles, faces at different heights and lap joints are unsupported. Previously trimmed flange outlines or legacy relief that removes bend width may prevent treatment. Relief defined on the parent plate can be combined.

Procedure

• Open Sheet metal, Corner treatment and choose the joint type.

• Choose Select in CAD beside Solid and select the sheet-metal part.

• Choose Select in CAD beside Adjacent flange faces. Select two broad side faces for Butt, or two or more coplanar rim faces for Miter.

• Enter the gap and choose Apply. Calculation runs at this point.

• Use Unfold and Refold if needed.

The diagram changes with the joint choice. It explains dimension locations; it is not a live geometric preview of the entered values.

Supplied example

The sample has eight solids and twenty text annotations. Row A shows Butt and row B shows Miter; each row contains Before, After, Unfolded and Refolded shapes. Dimensions are illustrative rather than manufacturing recommendations.

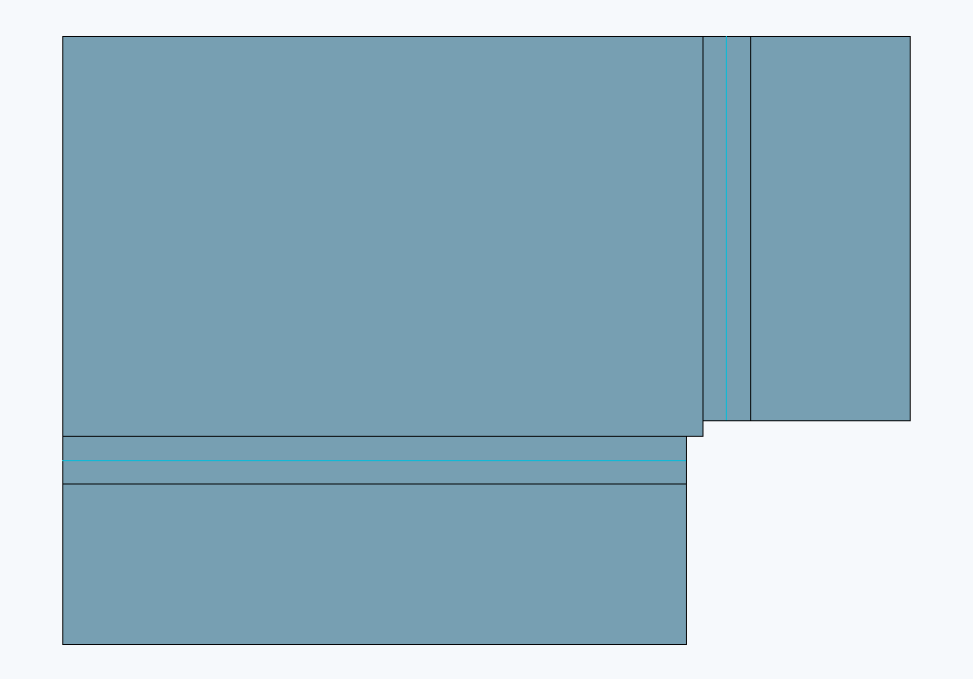


Figure 92 Unfolded butt example

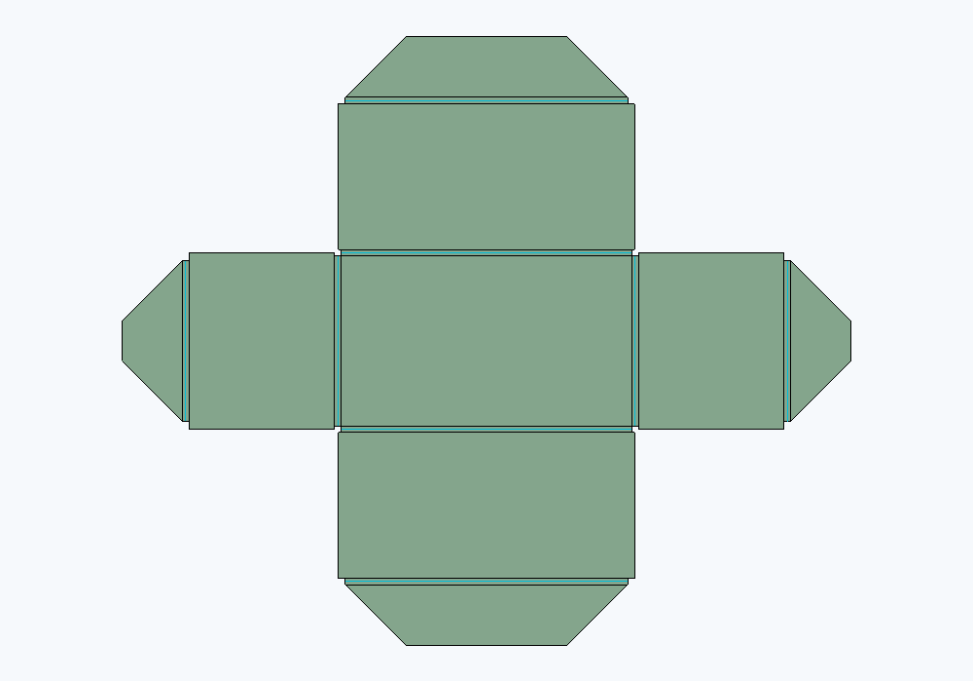


Figure 93 Unfolded miter example

Open the supplied corner example

Bend allowance table: CSV format and dimensions

Bend table imports a CSV into project defaults or an existing part. Use one material/tool combination per file, millimeters, degrees and decimal points. Columns:

```
Material,Tool,Thickness,Radius,Angle,Allowance
```

Allowance is the bend allowance. Thickness must match a table row. Radius and angle interpolate only within a complete table grid; no extrapolation occurs. Remove table returns to K-factor. Applying to an existing part retains the blank and adjusts the bend-band boundaries.

Partial unfold with stationary face

• Choose Unfold → Selected bends.

Sheet-metal help

A table stores measured bend allowances for a material/tool combination, indexed by thickness, inside bend radius and bend angle. It replaces the K-factor calculation for the applicable conditions.

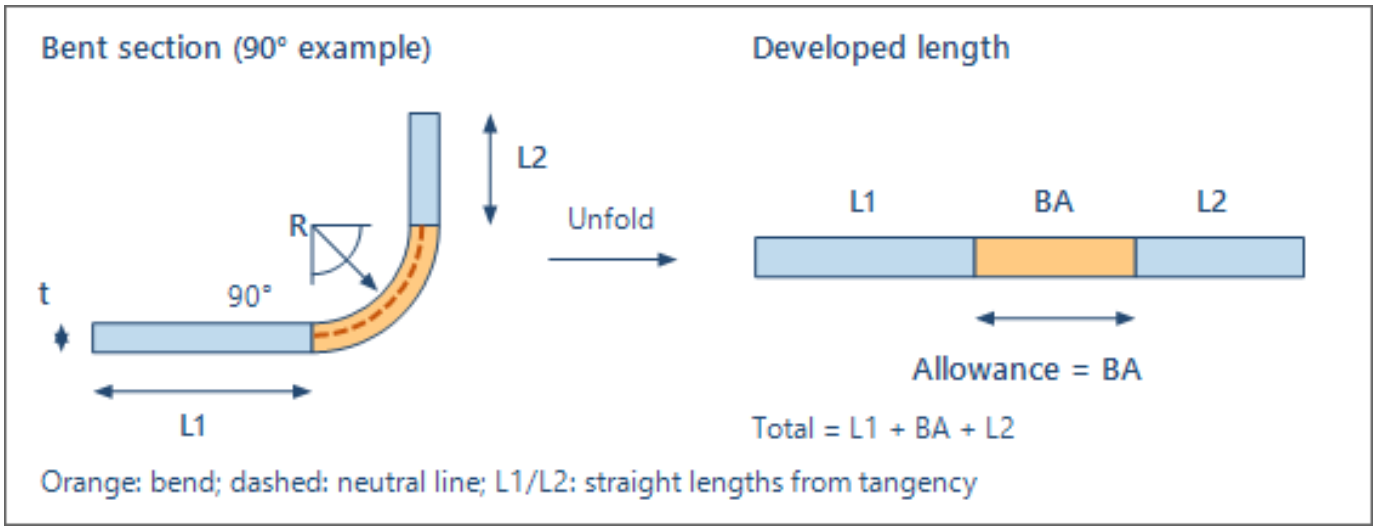


Figure 94 The orange bend band develops into BA. Total length equals the two straight tangent lengths plus BA.

L1 and L2 are straight lengths measured from the bend tangencies, not outside dimensions measured to the virtual sharp corner. Allowance is BA: the developed length of the bend band.

1. CSV format

Use these six headers in this order:

```
Material,Tool,Thickness,Radius,Angle,Allowance
DEMO,DEMO,1,1,45,1.099557
DEMO,DEMO,1,1,90,2.199115
DEMO,DEMO,1,1,135,3.298672
```

Save the sample CSV

Calculated example only: thickness 1 mm, inside radius 1 mm, K=0.4. These are not measured manufacturing values. Replace them with measurements for your material and tooling.

Column	Meaning
Material	Material identifier; same on every row in the file.
Tool	Tool/process identifier; same on every row.
Thickness	Positive thickness, mm.
Radius	Positive inside bend radius, mm.
Angle	Bend rotation from flat, degrees; greater than 0 and at most 180.
Allowance	Positive developed bend length BA, mm.

In Excel, fill columns A–F and save as CSV UTF-8 (comma delimited). Use decimal points and numeric values without unit suffixes. One file contains one material/tool combination. Names identify the loaded table; they do not automatically select other tables.

Angle is the rotation from flat, not the included opening angle. An included opening of 135° means Angle=45°. An opening of 45° means Angle=135°. Use positive table angles for either bend direction.

2. Bend allowance versus bend deduction

If your shop's value is subtracted from the sum of outside dimensions, it is bend deduction BD and must be converted before entering Allowance.

For t=1, R=1 and BD=1.800885 mm, enter BA=2.199115 mm. Outside dimensions A=20 and B=30 give a flat length of 48.199115 mm. These outside dimensions extend to the virtual sharp corner.

For bend rotation θ in degrees, BA = 2 × (R+t) × tan(θ/2) − BD applies for 0<θ<180°. Do not use this conversion for a 180° hem or for deductions based on other dimension references.

The K-factor relation is BA = θ × π / 180 × (R + K × t). An active table takes precedence.

3. Import and scope

• Open Sheet metal → Bend allowance table.

• Select the scope. New-part defaults apply to future sheet-metal parts in this document. They do not update all existing parts. Selected part requires one solid picked in CAD.

• Choose Import CSV, then Apply. The file picker opens when Apply is pressed.

• Check the completion message. Use Remove table and Apply to return the chosen scope to its configured K-factor.

Applying a table to an existing part retains its actual developed blank outline and changes the bend-band boundaries. It does not hold finished outside dimensions fixed while changing the blank size.

Refold partial developments before applying a table. Rolled parts use K-factor and do not accept this press-bend table.

4. Coverage and interpolation

• Thickness must match; there is no interpolation between thicknesses.

• Radius and angle interpolate only within the registered range. The sample covers R=1 and angles 45°–135°, not R=2 or Angle=30°.

• Interpolating both radius and angle requires all four surrounding combinations.

• Duplicate thickness/radius/angle combinations are invalid. Out-of-range requests and missing interpolation corners fail when used.

• The accepted allowance range is θ×π/180×R ≤ BA ≤ θ×π/180×(R+t), equivalent to K between 0 and 1. Check the dimension definitions if measurements fall outside it.

Edit sheet metal with regular commands

• Select a stationary plane and one or more cylindrical bend faces.

• Use ordinary cuts, including slots crossing the open bend bands.

• Refold to return the bends, keeping the stationary panel in place.

Full development retains panel boundaries and bend center edges. Click a selected item again to deselect it.

Multi-edge flanges, open hems, jogs and contour flanges

Full-edge flange mode accepts multiple edges. Range mode accepts one edge. All selected flanges commit together only if every step succeeds.

Roll / loft

Smooth coaxial rolls and triangulated round–round, rectangle–round and rectangle–rectangle transitions are available. Set offsets, seam gap and segments; unfold, refold and edit dimensions. Triangulated dimensions locate the neutral surface and exclude finite brake radii and allowances. See instructions, datums and limits.

Manufacturing checks

• Inventor: punch tools

• Creo: forming and flat representation

Machined bends, automatic miters and dimension references

Change box width or depth with Move Faces

On an orthogonal sheet-metal box, moving a side wall also moves its attached rim and stretches the floor and adjacent walls. Thickness, bend radii, wall height, rim depth and 45° miters retain their sizes.

• Start regular Move Faces while folded.

• Select a broad side-wall face, inside or outside, or the broad face of its attached top flange. Both faces on that same side may be selected together.

• Move perpendicular to that wall, parallel to the floor. To change width from 100 mm to 120 mm, move one side outward by 20 mm.

• Commit to stretch the connected panels while leaving the opposite side fixed.

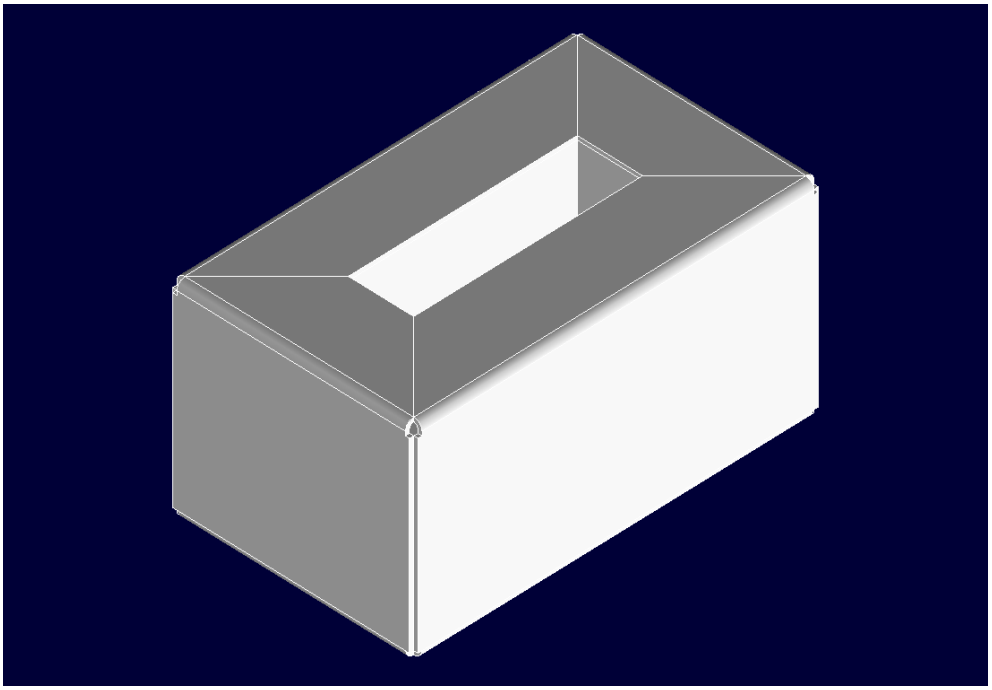


Figure 95 Box before editing, with four mitered rim flanges

Sample 19: width 100 mm, depth 60 mm.

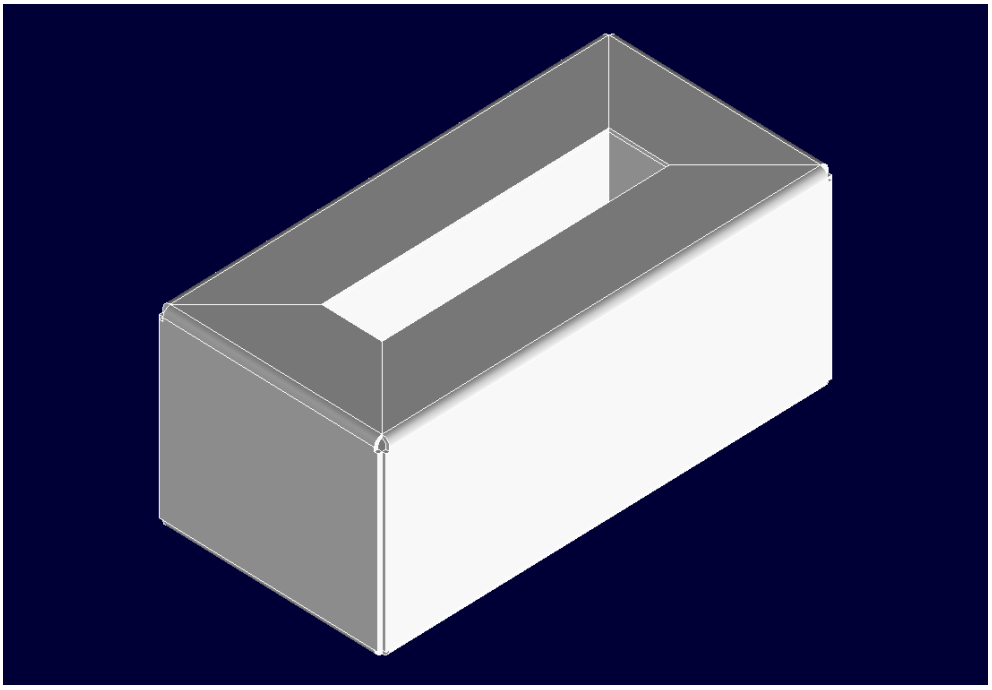


Figure 96 Box widened to 120 mm, retaining miter joints

One side moved by 20 mm: width 120 mm, same depth and height.

The cyan preview shows only the destination of selected faces. The connected body is rebuilt at commit; separated preview faces do not mean the committed sheet will be disconnected.

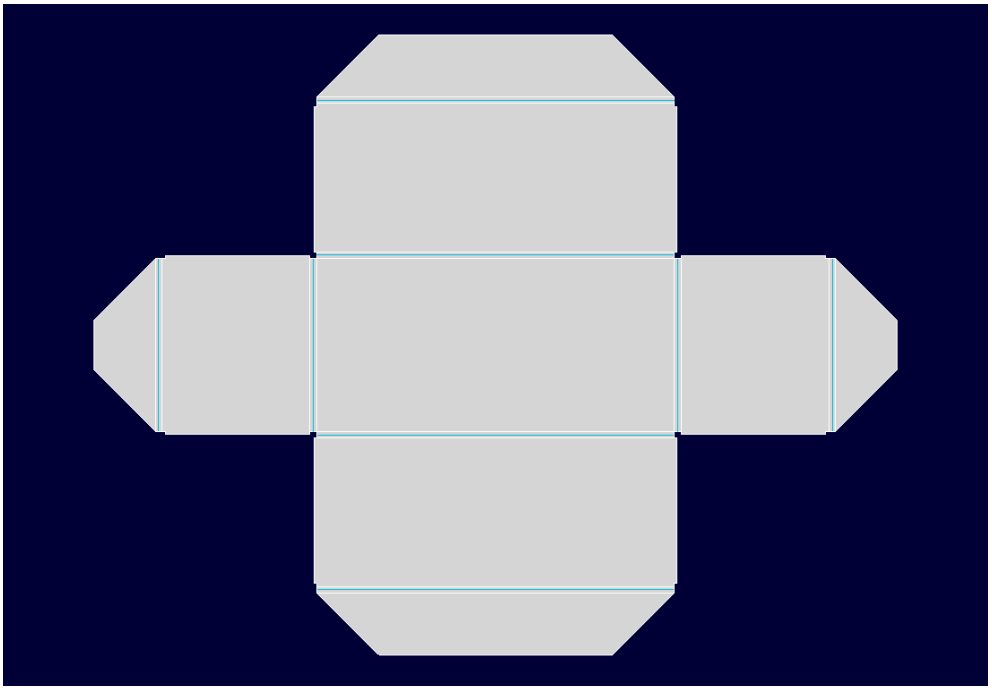


Figure 97 Unfolded widened box with bend center lines

The resized blank supports Refold, save/reload and Undo/Redo.

This applies to faces on one side of an orthogonal box with continuous straight ends. Shortening is supported unless material crosses. Ambiguous bays or split bend regions are rejected. Holes keep their sizes: holes on the moving wall follow it, while holes in the floor and adjacent walls remain in place.

Selecting a narrow end face instead performs the individual end edit described below.

Use unfolded bend lines as drafting references

Unfold now adds a cyan center line within each bend band, halfway along its developed allowance including the K factor. Lines are created on both broad skins.

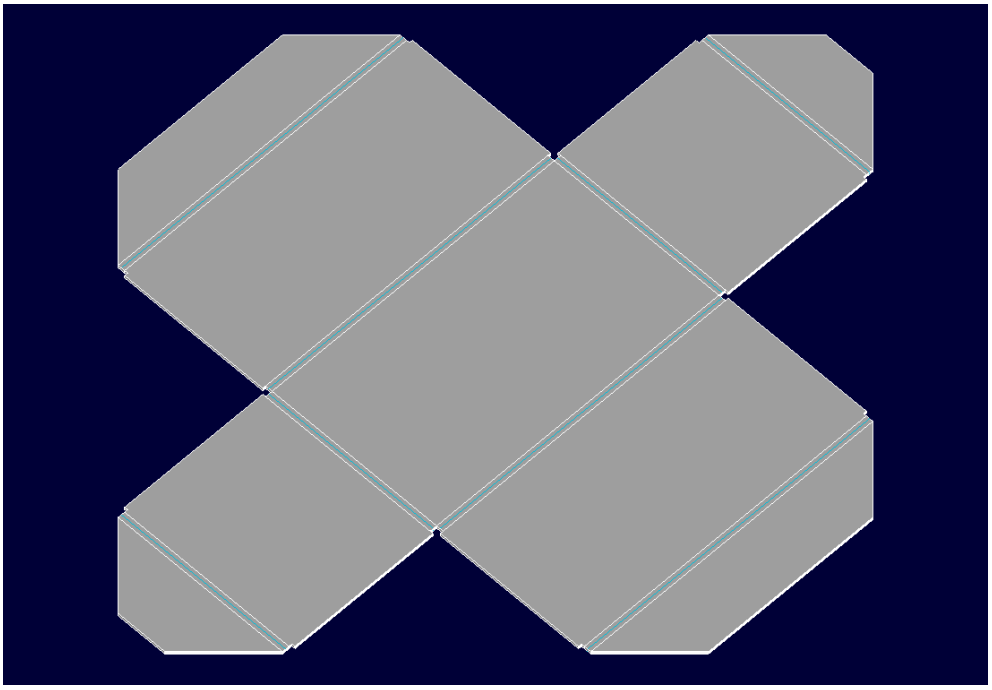


Figure 98 Cyan bend centers between white bend-band boundaries on an unfolded box

White: region boundaries. Cyan: bend-band centers. Zoom in to distinguish all three lines.

• Unfold. For a blank saved by an older version, Refold and Unfold again.

• Align the work plane to the plate surface when drafting on that surface.

• Select a cyan line as the reference for a regular parallel draft line, or use endpoint, midpoint and on-line snaps.

• Draw the required outline or slot, make the through-cut, then Refold.

These are real selectable edges, not cuts: thickness and volume remain unchanged, with one solid. Holes and notches interrupt the lines. They follow solid moves/rotations, save/reload and Undo/Redo, and disappear when refolded. Draft lines you create remain normal independent objects. Flat-pattern output does not export these seams as extra cutting lines.

Registered sheet-metal solids support regular Move faces, Offset faces and Taper faces, updating both geometry and developed material. These images are native CAD results from a copy of sample 17.

1. Resize the flange width with a fixed bottom

• Start Move faces or Offset faces while folded.

• Select the narrow vertical wall end (red), or the cross-section at the bend end (yellow). Do not select the broad wall or cylindrical bend surface.

• Move parallel to the bend axis. For Offset faces, a negative value shortens the end and a positive value extends it.

• Red only moves the wall end; yellow only moves the bend end. The unselected end and planar bottom stay fixed. This changes the setback between those ends and adjusts the root notch.

• To preserve the setback, select both ends. Each moves once by the same amount.

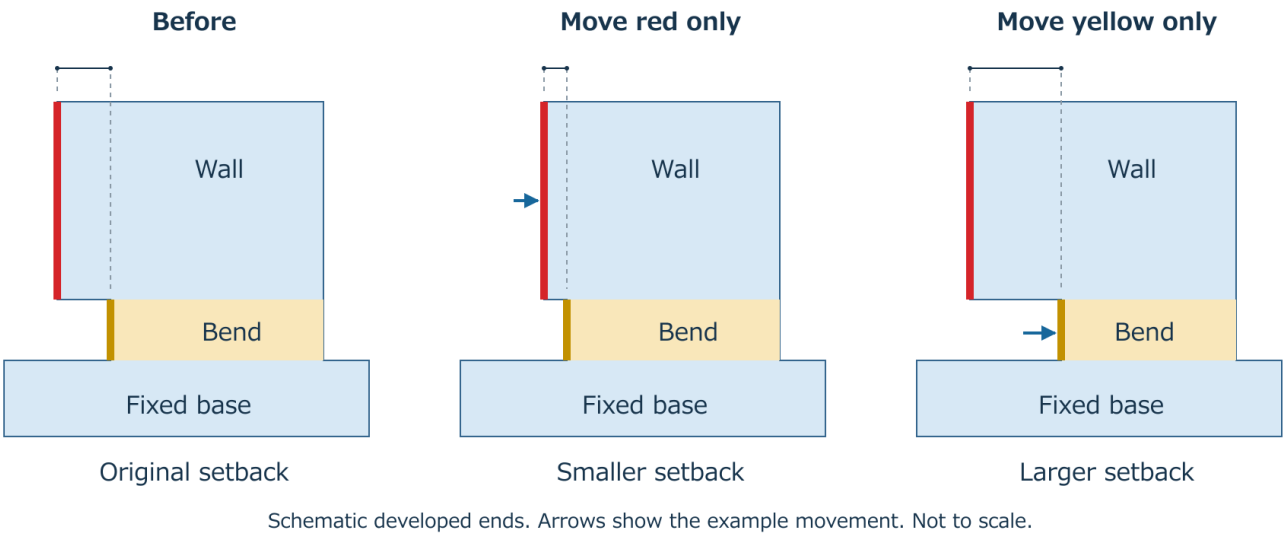


Figure 99 Separate wall-end and bend-end edits change their setback.

Adjust the relative setback independently of the full flange width. A physical face merged across multiple regions selects the regions it includes.

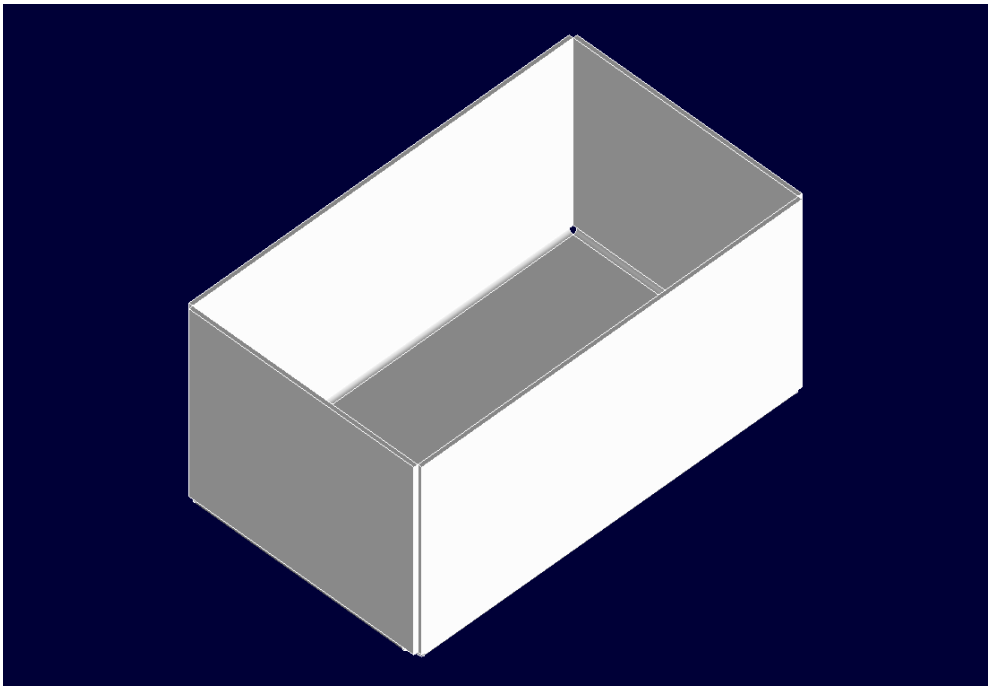


Figure 100 Shortened bend end with the wall end and bottom unchanged.

Moving only the bend end inward widens the root notch.

Relief dimensions are editable in other CAD systems: SOLIDWORKS exposes relief width and depth; Fusion exposes width, depth and remnant. Their general Move Face commands need not behave identically.

Extensions that collide with another panel and edits that disconnect material are rejected.

2. Edit the unfolded blank

• Unfold.

• Use regular face editing on the narrow edge faces, moving within the sheet plane. Outline taper with unchanged thickness is also supported.

• For a slot, create a cutter through the full thickness from a closed slot profile and use regular Boolean subtraction. The cut may cross bend bands.

• Refold to carry the edited material into the base, bends and flanges.



Figure 101 An obround slot across the bend region in the flat blank.

Unfolded: through-cut slot.

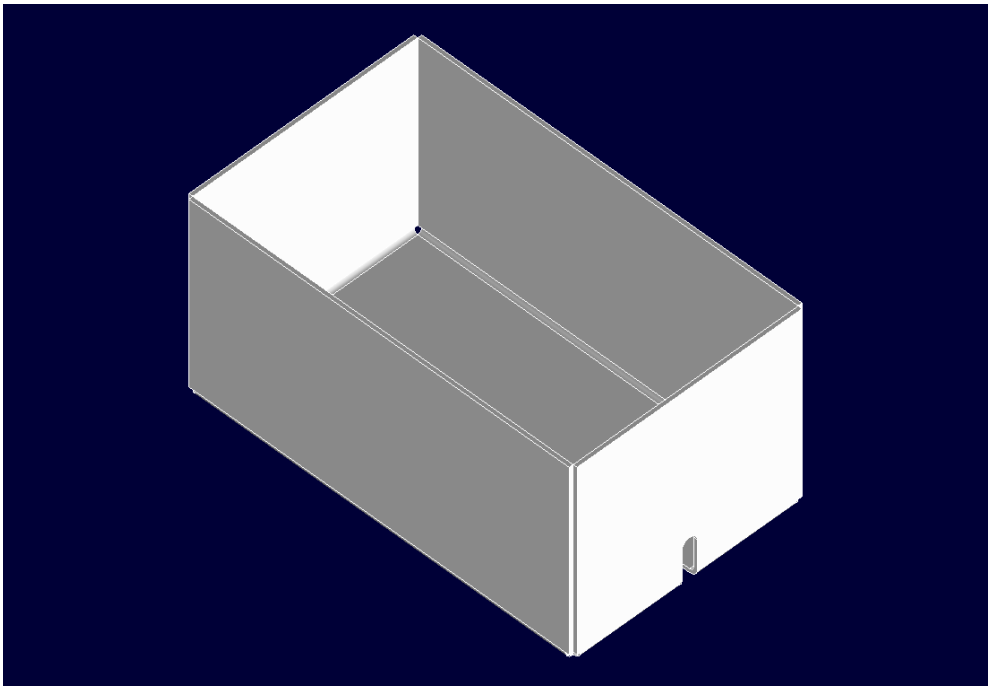


Figure 102 The same slot through the folded root.

Refolded: the slot remains continuous.

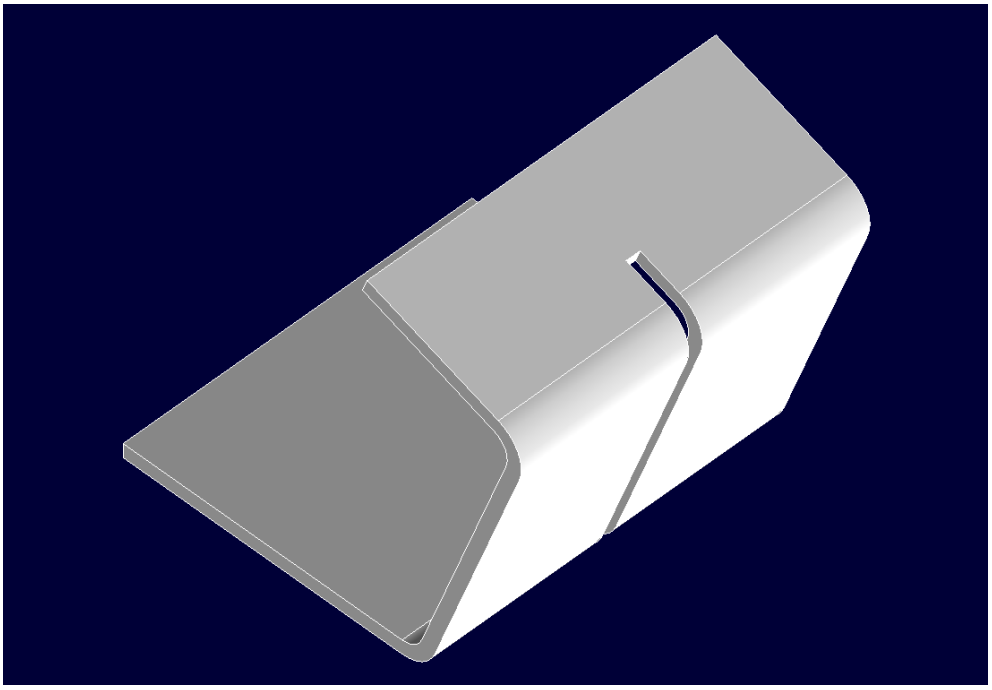


Figure 103 A slot crossing two nested bends.

Cuts can span multiple bends.

A drawn curve alone does not cut the solid. Added outline material must keep thickness and connectivity and have an unambiguous panel or bend owner. Separate additions and overlapping ownership are rejected.

3. Make a trapezoidal flange

• Use regular Taper faces on a narrow side end. Both ends can be selected together.

• While folded, use the long straight tangent between the planar flange and bend R. While unfolded, the flange-to-band seam or the short thickness edge at the root is also available.

• +45° narrows the tips; −45° widens them. The selected flange outline changes while keeping the thickness and root.

Illustrated guide: taper box return flanges while folded, or use retained edges while unfolded.

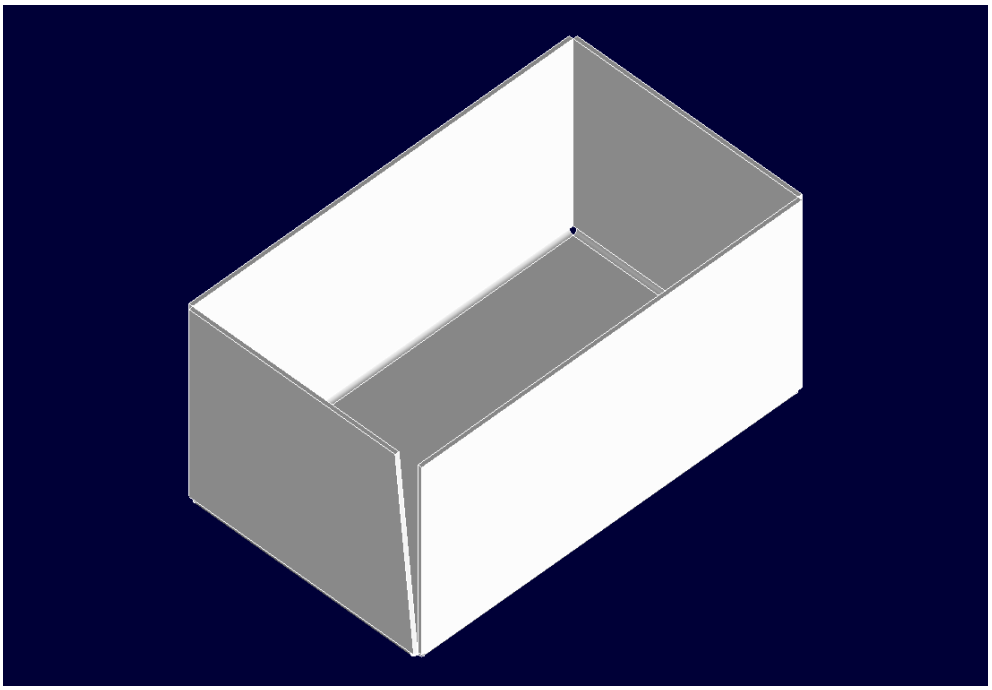


Figure 104 A tapered flange outline in the folded box.

Folded: tapered end.

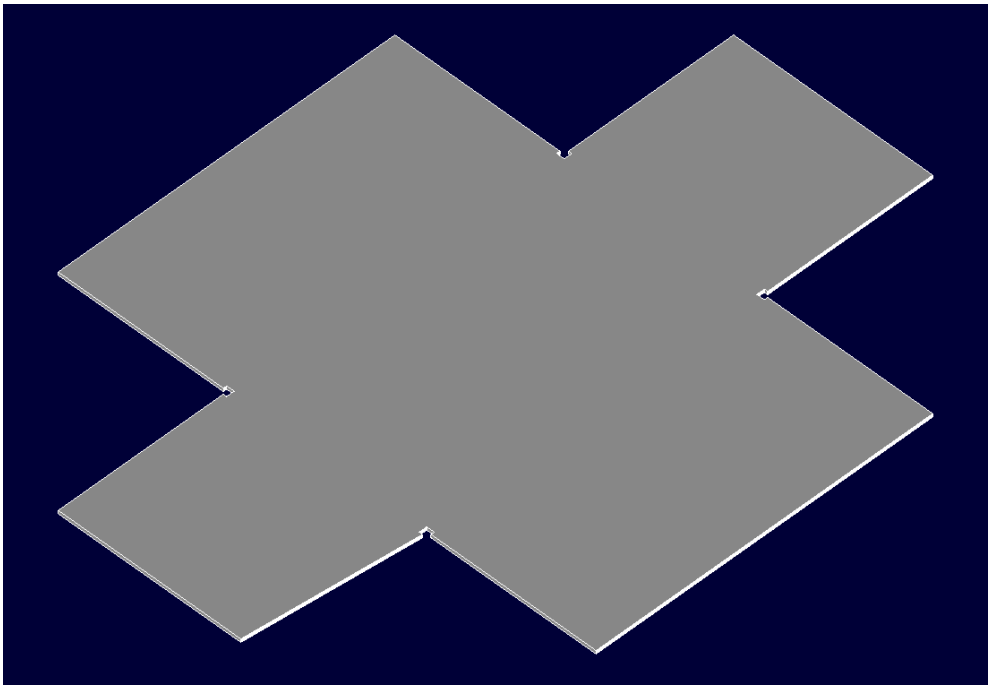


Figure 105 The trapezoid retained in the flat blank.

Unfolded: the edited outline remains.

This is outline taper. Tilting the broad faces to make thickness decrease toward the tip is unsupported.

4. Fill a corner step with regular extrusion while folded

• Extrude and unite the narrow end face at the step. Keep both broad skins at their original positions so sheet thickness stays constant.

• Run Unfold to incorporate the planar addition or removal into the developed outline.

• Refold to recover the edited shape. A JDC saved after extrusion can also be unfolded.

Verified on sample24 with 0.5 mm and 1 mm corner fills, including several filled corners. Through-cuts confined to planar panels are supported. Raised bosses, blind pockets, edits to bend R and ambiguous bridges between panels cannot be reconciled. Make slots crossing bend bands in the unfolded state.

Persistence and limits

Face commands update sheet data and geometry in one Undo/Redo step. JDC save/reload retains the result. Heavy reconstruction runs at commit, not during selection or value entry.

• Blind pockets, thickness changes, disconnected material and folded collisions are rejected.

• Angle/radius edits and flange addition retain machined bend regions. Straight length is calculated to preserve the developed outline. Regenerate reliefs or automatic corners before individual machining.

• Planar changes from regular extrusion, union or subtraction are incorporated at the next Unfold. Thickness and bends must remain unchanged, and the reconstructed exterior must match the complete edited shape.

Partial unfold with a stationary face and additional regular edits

Narrow or widen flange ends by 45 degrees

To build all four top flanges, use Edge flange with 45-degree ends at creation time. The worked example includes the range setbacks needed to avoid overlapping bend bands.

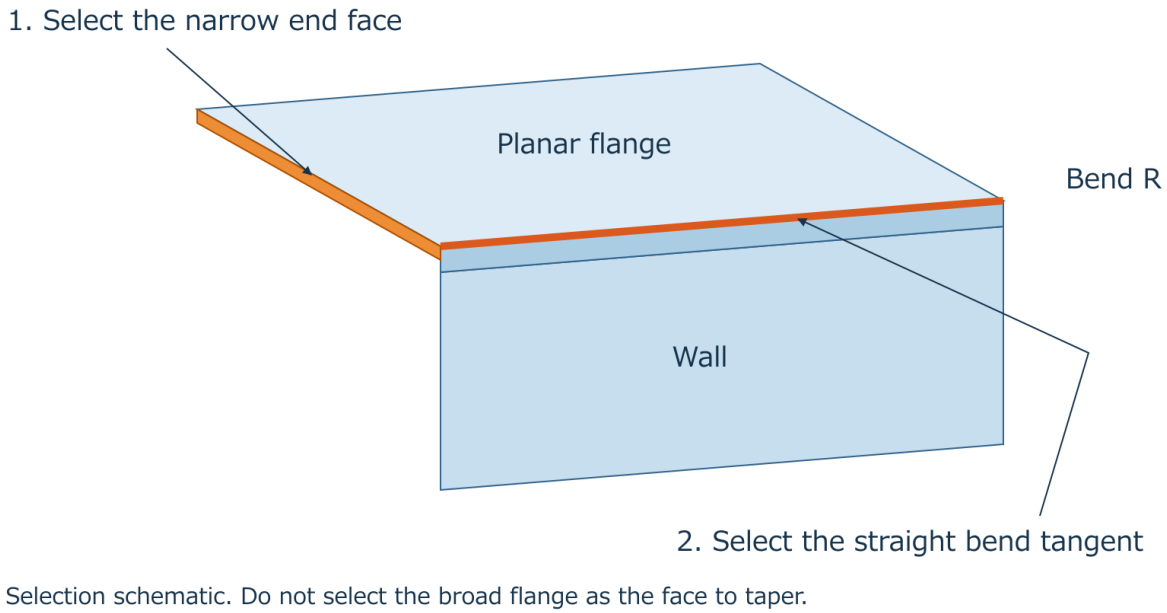
Use regular Taper faces to narrow or widen flange ends while folded. The examples use a copy of sample 18, including an outward-facing return flange.

Edit while folded

• Select the narrow thickness end face in Taper faces. Select both ends to apply the same angle to both.

• For Reference boundary edge, select the long straight tangent between the planar flange and its bend. Do not select the bend-end arc or the free tip edge.

• Commit +45° to narrow the tips or −45° to widen them. The root and sheet thickness stay fixed.



Selection schematic. Do not select the broad flange as the face to taper.

Figure 106 Select the narrow end, then the flange-to-bend tangent.

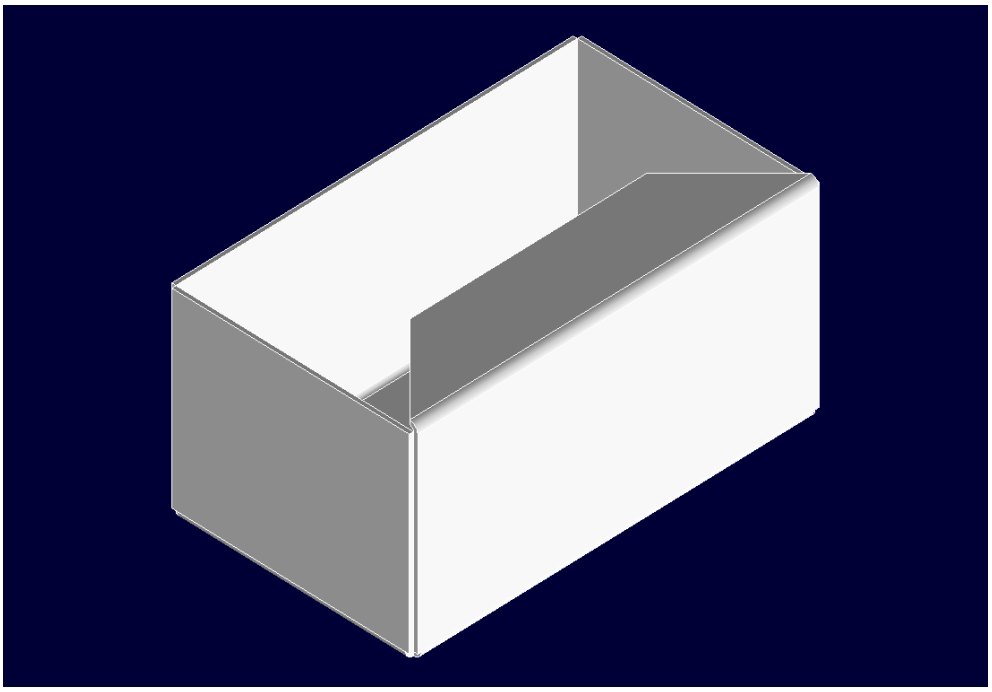


Figure 107 Both flange ends narrowed by 45 degrees.

+45°: narrower tips.

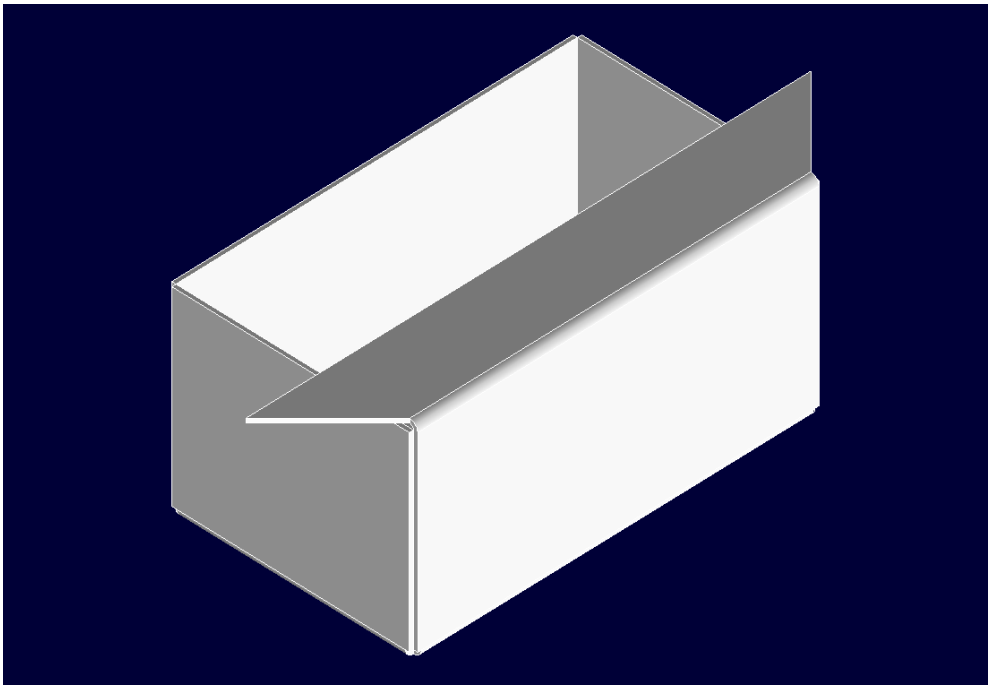


Figure 108 Both flange ends widened by 45 degrees.

−45°: added material widens the tips.

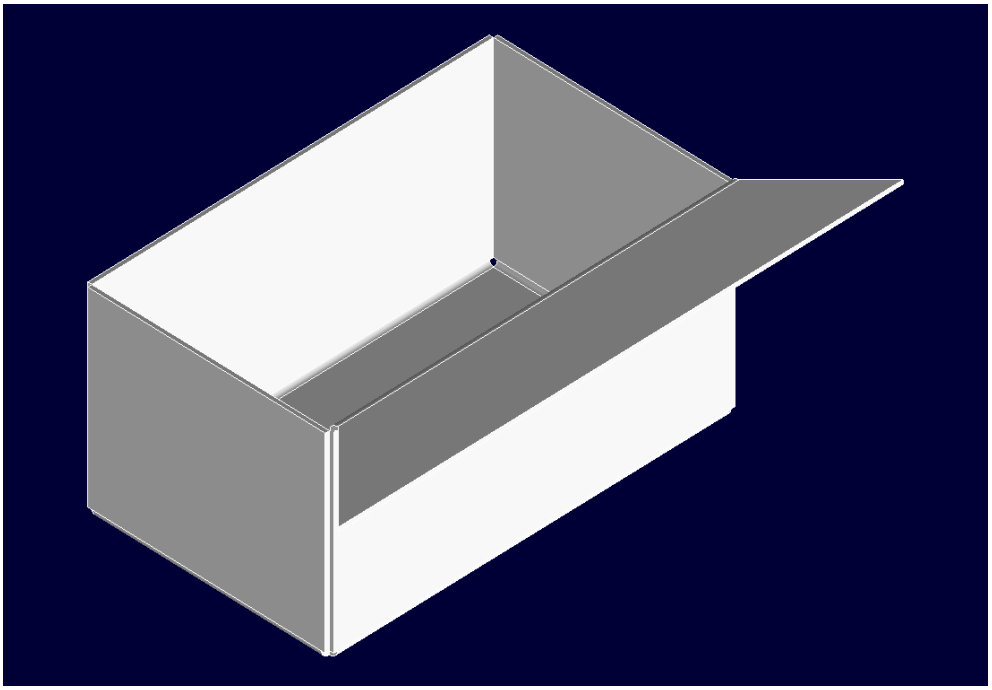


Figure 109 An outward-facing return flange with widened ends.

The same operation works on an outward-facing flange. Sign means narrowing/widening away from the root, not screen direction.

Edit the unfolded blank

- Unfold. Base, wall, bend band and return flange retain real face and edge boundaries.
- Select the narrow flange end. Use the flange-to-band seam or its short thickness edge as the taper reference.
- Commit +45° or −45°. Regular Move faces and Offset faces can also resize the selected region.
- Refold. Added and removed outline material is retained.

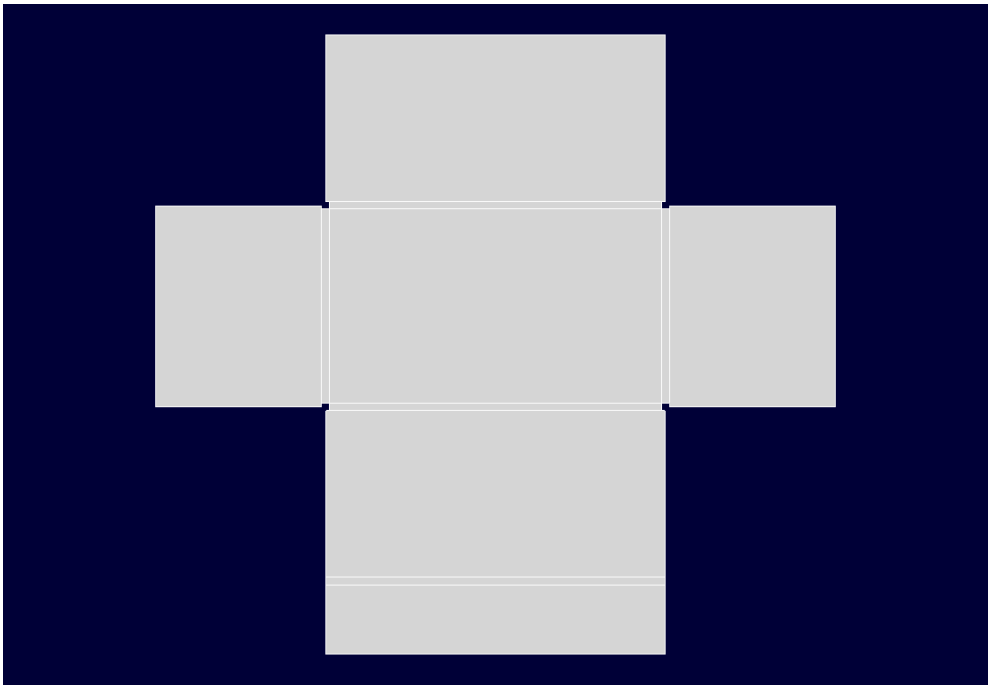


Figure 110 A blank retaining face boundaries on both sides of its bend bands.

The two lines bound a bend band. They do not split the part into separate solids.

For a blank saved by an older version, Refold and then Unfold to create the seams. Flat-pattern export still uses the outer/hole contours as CUT lines; these editing seams are not extra cutting lines.

Scope

• This changes the planar flange outline while keeping the root and bend R. To cut diagonally through the R as well, unfold and make a through-cut across the bend band.

• This does not automatically miter adjacent flanges or calculate their clearance. Edit their ends and check contact, overlap and the required gap. Creating and auto-mitering all four top flanges in one operation is not yet supported.

• Thickness changes, disconnected material and material overlaps are rejected. Arbitrary folded-state Boolean edits are not tracked.

• Save/reload and Undo/Redo retain the result. Machined bend bands still restrict subsequent angle/radius changes.

Machined bends, automatic miters and dimension references

All geometry is computed on Apply. A successful result replaces the solid atomically and supports Undo/Redo.

Edit bends without removing slots or notches

• Unfold and make through cuts using ordinary modeling commands, including cuts across bend bands.

• Refold.

• In Edit bend / flange, select the cylindrical bend face and change angle or inside radius.

The developed outline and hole positions remain fixed. The actual material is repartitioned between the bend band and planar panel; straight length is calculated. You can also add a flange at a free straight edge while retaining older machined bends.

Thickness changes, disconnected material, crossing another bend attachment and folded collisions are rejected. Regenerating bend reliefs or automatic miters after individual machining remains unsupported; ordinary outline edits remain available.

Automatic box-rim miters

• Select a straight wall-top edge in Edge flange.

• Choose Full edge or Start/end range, square ends and Automatic miters: Automatic.

• Set length, 90-degree angle, radius and gap (for example 0.2 mm), then Apply.

• Repeat on adjoining walls. Both flange ends at each detected corner are adjusted. The first flange has square ends until its neighbors exist.

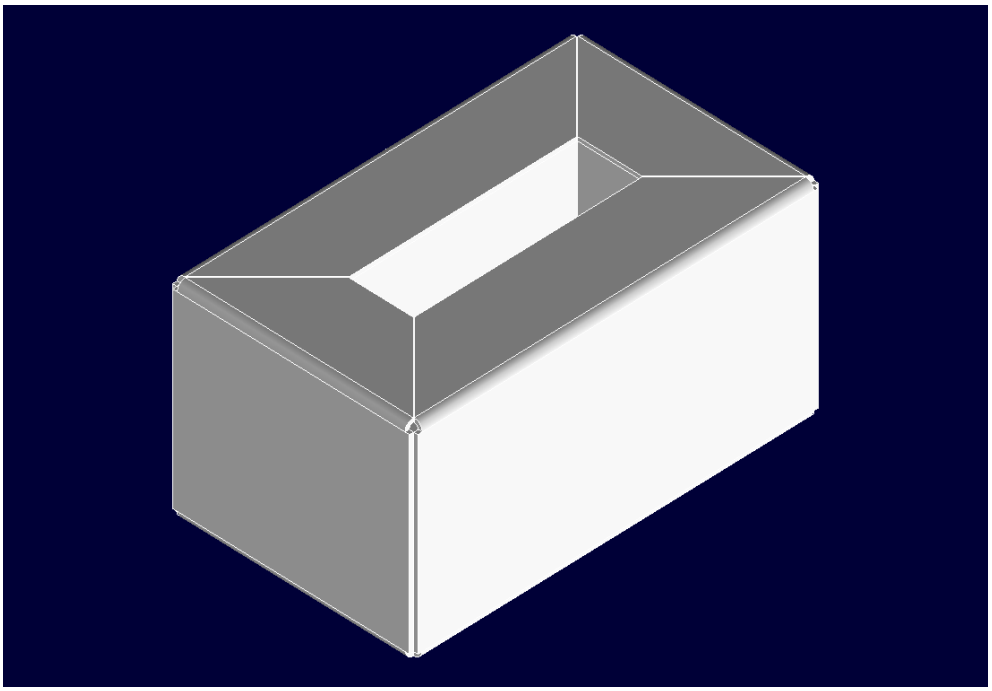


Figure 111 Automatically mitered inward rim

Inward: narrow the tips.

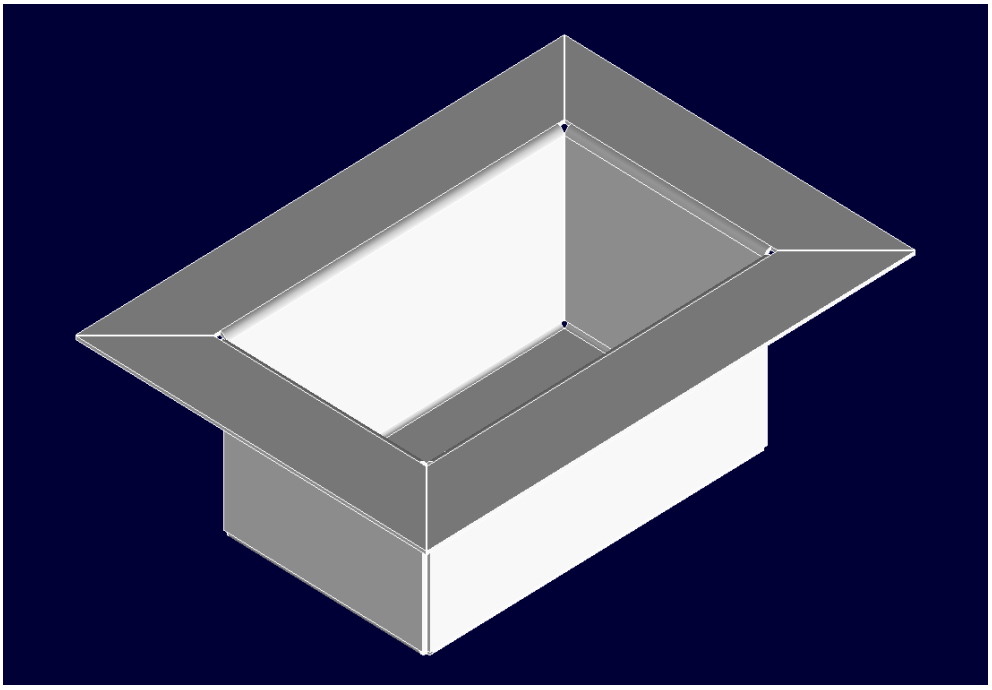


Figure 112 Automatically extended outward rim

Outward: extend where needed. Pick the outer wall-top edge.

The gap is measured perpendicular to the planar seam. Bend ends are set back as needed; their curved opening is not the same measurement as the planar gap.

Leave margins at the ends

For a 57.9 mm edge with 1 mm removed at both ends, choose Start/end range and enter start 1 mm, end 56.9 mm. Keep square ends and enable Automatic miters. Rectangular or round reliefs generated in the supporting sheet are retained while adjoining rims are mitered.

End margins and the diagonal seam gap are separate dimensions. For a 1 mm seam, also enter Gap = 1.0000 mm. A gap setting of 0.1000 mm still produces a 0.1 mm seam regardless of the end margins. Creation, unfolding and refolding with 1 mm end margins and 0.1 mm or 1 mm seams have been verified for inward and outward rims. A 19.1 mm flange length and adding a partial flange beside an existing full-width rim are also supported. The selected range limits the bend band width. Bend ends can be set back further for the seam, and outward planar panels extend where needed.

Change the gap later

Use Corner → Miter (coplanar flanges), select two or more broad adjoining flange faces and enter a gap. Select all four rim faces to adjust all corners. Both increasing and decreasing gap are supported.

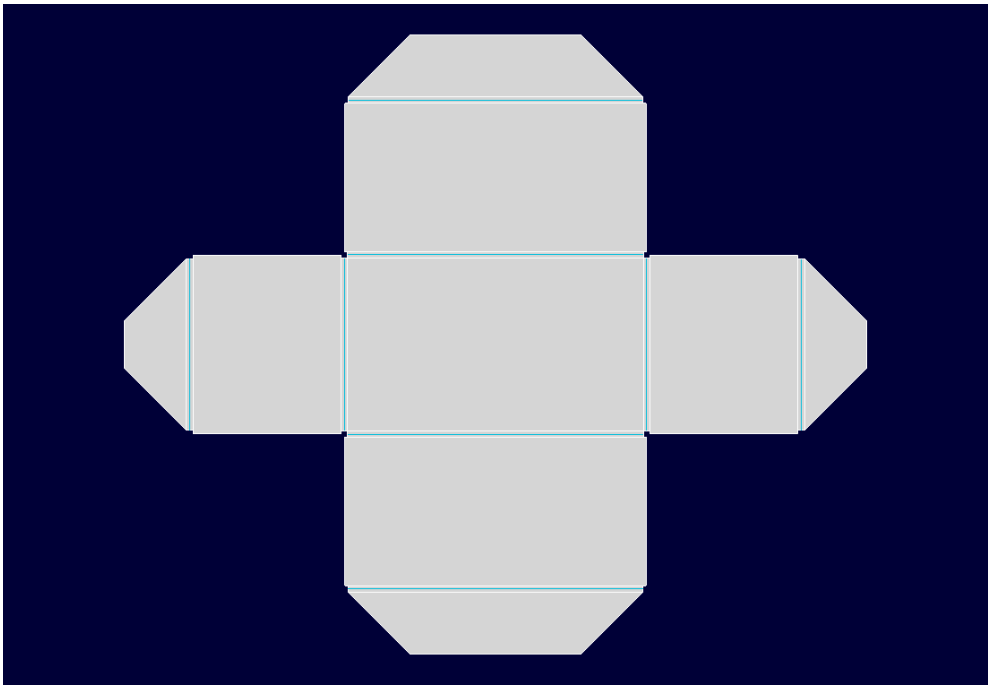


Figure 113 Unfolded automatically mitered box

Miters, region boundaries and bend center lines remain in the unfolded model.

Supported joints have coplanar, perpendicular neighboring flanges. Parent-sheet reliefs generated for a partial range are supported. Individually machined outlines and legacy reliefs that trim the moving bend/panel width remain unsupported. Arbitrary angles, stepped planes and automatic overlap joints are outside this mode. Butt retains the original end-setback operation.

Alignment and length reference

Bend start (tangent) and Straight after bend preserve the previous interpretation. Existing files retain their dimensions. Focusing either reference field shows the section diagram.

Alignment to selected edge / line	Start setback
Bend start (tangent)	0
Inside virtual sharp	R × tan(angle/2)
Outside virtual sharp	(R + thickness) × tan(angle/2)
Developed bend center	Half the bend allowance

Virtual sharp means the intersection of the extended planar sheets. Edge flange consumes the required strip of supporting material; missing material, holes or another bend in that strip are rejected. Bend along line supports the same alignment choices.

Length reference	Straight length L
Straight after bend	Entered dimension
Inside virtual sharp	Dimension − R × tan(angle/2)
Outside virtual sharp	Dimension − (R + thickness) × tan(angle/2)

With thickness 2 mm, inside radius 3 mm and angle 90°, an outside dimension of 20 mm gives a straight length of 15 mm. The length reference is saved and restored when editing an ordinary flange. Machined blanks show calculated straight length instead. Alignment is a creation-time setting; later angle/radius edits keep the bend start fixed. At other angles the dimension runs along the flange, not vertically on the screen. Nonpositive straight lengths are rejected.

Roll / loft: round and rectangular transitions

Check accepts a minimum inner radius and minimum hole clearance to free edges and bends. Zero disables the limit. Holes within a bend band also fail clearance. Use your material/tool limits. Geometry checks are not predictions of cracking, thinning, springback or tool access.

Official design references

Open Sheet metal → Roll / loft (STD / PRO). Round-to-round, rounded rectangle-to-round and rectangle-to-rectangle offer Smooth forming or Triangulated (facets). The faceted method preserves triangle edge lengths and does not model finite press-brake radii or allowances.

Automatic or specified seam position

Roll / loft offers Seam position → Automatic / Specified for smooth and faceted round-to-round, rectangle-to-round and rectangle-to-rectangle transitions, and cylindrical/conical rolls. Automatic retains the previous location and is the default for existing files.

• Enter section dimensions or select both section sketches.

• Choose Specified, then enter Seam direction (deg): 0° = local +X, 90° = +Y, 180° = −X, 270° = −Y, viewed from above the lower profile.

• Alternatively, press Pick seam direction and pick a point on the lower profile outline. The angle is measured from the lower profile center, not the center of a corner arc. The center itself cannot define a direction.

• Apply, then Unfold to inspect the blank. Edit existing roll exposes the same fields.

Dimension input starts with world X/Y axes. Sketch input uses the lower profile axes; moving or rotating a part retains its local axes. Pick in CAD when these differ from screen directions. The seam runs from the selected lower-profile position along the corresponding generator; the complete part is not rotated.

Specified mode	Gap convention
Rectangle-to-round, rectangle-to-rectangle and faceted lofts	Width perpendicular to the seam on the developed sheet, removing half on each side. A seam on a rectangle-to-round planar side opens at the upper tip as well.
Smooth round-to-round	Neutral endpoint distance at the smaller circle, scaling with radius at the other end.
Cylindrical/conical roll	The roll angle controls the opening; Seam direction locates the center of the missing angular sector.

Existing holes and notches retain their size and developed coordinates. Large seam-position changes also change the developed outline; an edit is rejected if the machining cannot fit the new material. Geometry is calculated only on Apply.

Smooth round-to-round forming

Select Round → round, then Forming method → Smooth forming (roll / die). Enter the neutral radii, height, X/Y center offsets, thickness and K factor. X = Y = 0 gives concentric profiles. No facet count is needed.

The body uses continuous surfaces at constant normal thickness. A concentric cone develops into two exact circular arcs and two straight sides; a concentric cylinder into a rectangle. Eccentric cones and cylinders develop into noncircular spline outlines, with a calculation tolerance of 0.00001 mm. This is a numerical tolerance, not a forming-process accuracy. DXF and DWG preserve the arcs/splines on SM_CUT without internal fold guides.

Seam gap is the neutral endpoint distance at the smaller circle. Generator-aligned seam sides keep the surface developable, so the gap at the larger circle scales with radius. Radii and height locate the neutral surface; skins are normal offsets by −K×t and +(1−K)×t. In eccentric parts, skin boundaries are not circles in parallel XY planes. Use the original smooth cylinder/cone mode when concentric inner-radius dimensions are required.

Existing unmachined round-to-round lofts can switch forming method through Edit existing roll. The two methods use different seam-gap conventions. See the rectangle-to-rectangle instructions below. Through holes and notches made while unfolded can be refolded; see Machining below.

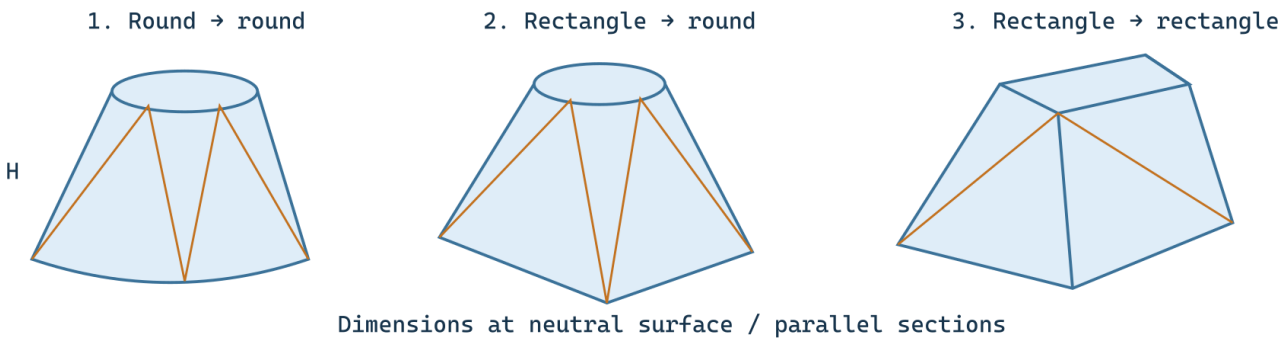


Figure 114 Three loft section families and neutral dimensions

Smooth rounded-rectangle to circle

• Choose Enter dimensions, Rectangle to round (corner R / eccentric), and Smooth forming.

• Enter rectangle width, depth, corner R, circle radius, height, and X/Y offsets. Corner R must be positive and less than half both rectangle dimensions.

• Set thickness, K factor and seam gap, then Apply. Use Sheet metal unfold / refold.

Four planar sides join four curved corner regions tangentially. Eccentric profiles, either size of circular end, and equal corner/circle radii are supported. The developed boundary preserves neutral lengths and uses smooth splines and straight sides without triangulation guides. DXF and DWG retain those curves.

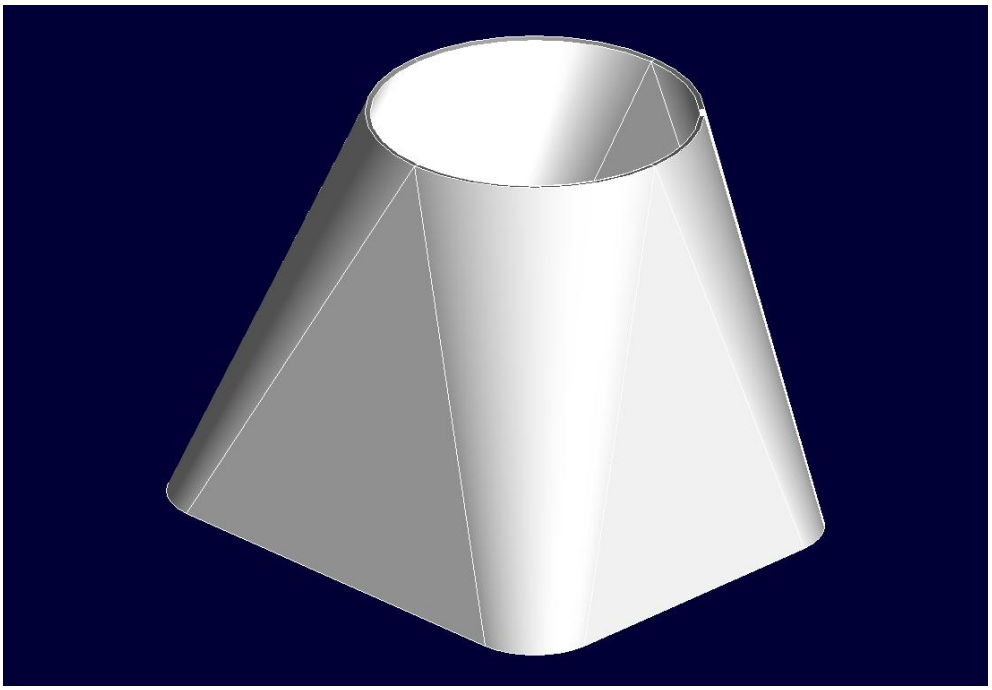


Figure 115 Actual eccentric rounded rectangular transition

100 × 80, corner R12 → circle R30; H90; offsets X20 / Y−10

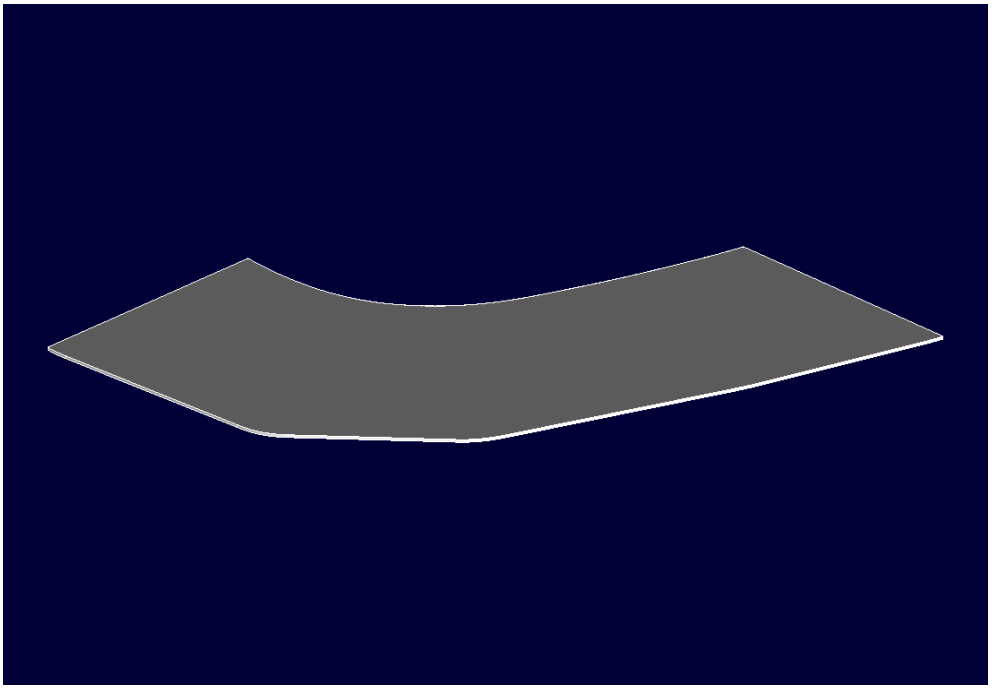


Figure 116 Actual developed blank

The same part unfolded.

Dimensions and sketches define the neutral surface. Corner R is a profile radius, not a minimum inner bend radius. Skins are normal offsets by −K×t and +(1−K)×t; their edges need not lie in the original sketch planes.

With Automatic seam position, the seam is centered on the +X/+Y rounded corner. Gap is the neutral endpoint distance at the smaller of the corner and circle radii; the other end scales proportionally. It must be smaller than both radii. See Specified seam position below for other locations and its gap convention. Inner/outer sketch datums are not included.

Corner R also works in Triangulated mode, where arcs are approximated with facets. R = 0 retains legacy sharp rectangular corners. Unmachined existing parts can change radius and forming method with Edit existing roll.

Smooth rounded rectangle to rectangle

• Choose Enter dimensions → Rectangle to rectangle (corner R / eccentric) → Smooth forming.

• Enter width, depth and corner R separately for the lower and upper rectangles, plus height. The two radii may differ.

• Set upper-center X/Y offsets, thickness, K factor and seam gap, then Apply.

• Use Sheet metal unfold / refold. Edit existing roll changes either corner radius and section dimensions, even while unfolded.

Curved corners meet planar sides tangentially. Straight rectangular ducts, expanding transitions and eccentric transitions are supported. Each R must be positive and less than half that rectangle's width and depth. All dimensions locate the neutral surface; the skins are normal thickness offsets.

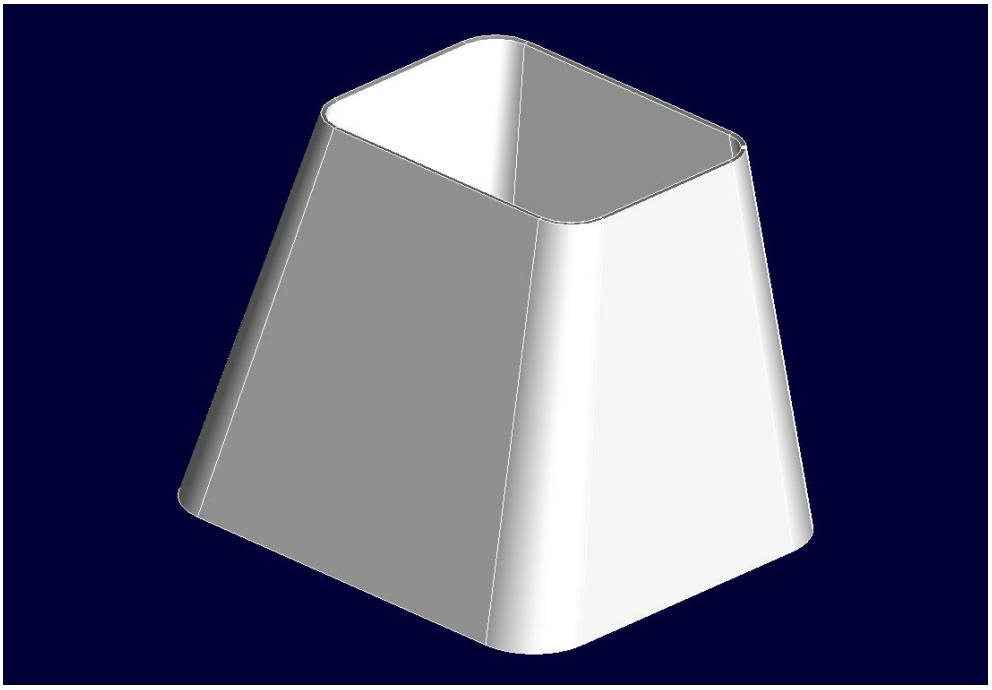


Figure 117 Actual eccentric rounded rectangular transition with two corner radii

Lower 100×80 R12 → upper 70×50 R8; H90; offsets X20 / Y−10

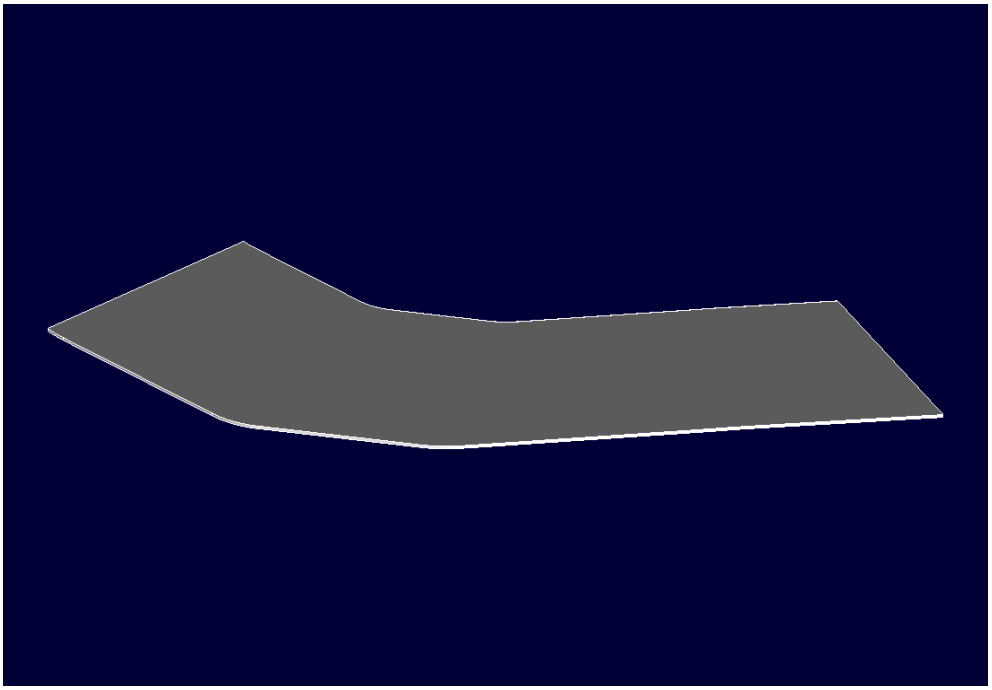


Figure 118 Actual unfolded rectangular transition

The same part unfolded, without internal triangulation guides.

For two rectangular sketches, keep planes parallel and corresponding sides parallel. Select profile 1, then profile 2. A rounded sketch supplies its own R. Sharp profile 1 uses Lower rectangle R, and sharp profile 2 uses Upper rectangle R. Input sketches are retained without a live associative link.

With Automatic seam position, the seam is at the +X/+Y corner midpoint in local coordinates. Gap is measured at the smaller R and scales at the other end; it must be smaller than both radii. Sketch input derives local axes from profile 1. Seam position can be specified as described below. Tilted planes and twisted profiles are unsupported.

Triangulated mode also supports both corner radii. Set both to zero for legacy sharp rectangles. The ordinary-machining restrictions below still apply.

Create from two sketches

• Draw a rectangle and a circle, two circles, or two rectangles in parallel planes. Align corresponding rectangle sides. The pair may have any rigid orientation.

• Choose Create → Section input → Select two sketches.

• Pick Profile 1 and Profile 2. For two rectangles, profile 1 defines the lower section. A connected contour can be selected by clicking one of its lines or arcs. Re-select the same entity to deselect.

• Set forming method, thickness, K factor and gap, then Apply. Section placement, spacing, sizes and eccentricity come from the sketches.

A rounded rectangle supplies its own R. A sharp rectangle uses the entered corner R; two circles ignore it. Input sketches are retained, with no live associative link after creation. Edit existing parts through their stored dimensions.

Rectangles require four lines, or four equal-radius 90° arcs and four tangent lines. Planes must be parallel. Arbitrary splines, nonparallel profiles and intermediate sections are unsupported. The machining restrictions below still apply.

Machine the flat, then refold

• Select the loft and use Sheet metal unfold.

• Use the ordinary Hole command or subtractive extrusion to make through holes, slots or edge notches. Cut through the full thickness.

• Select the machined solid and use Sheet metal refold. Slots crossing planar/curved boundaries follow the original surfaces.

• Unfold again to recover the machined blank. Further cuts, native save/reload, and DXF/DWG export retain the machining.

Supported: concentric/eccentric smooth round-to-round, rounded rectangle-to-round and rectangle-to-rectangle, faceted lofts, and cylinder/cone rolls. Holes follow the neutral-surface development; a circle in the flat need not remain a planar circle after forming.

Blind pockets, variable thickness, disconnected pieces, added material outside the original blank, and machining while folded are unsupported. Loft dimensions and corner radii can be edited after machining as described below. Failed refolding leaves the original flat solid intact.

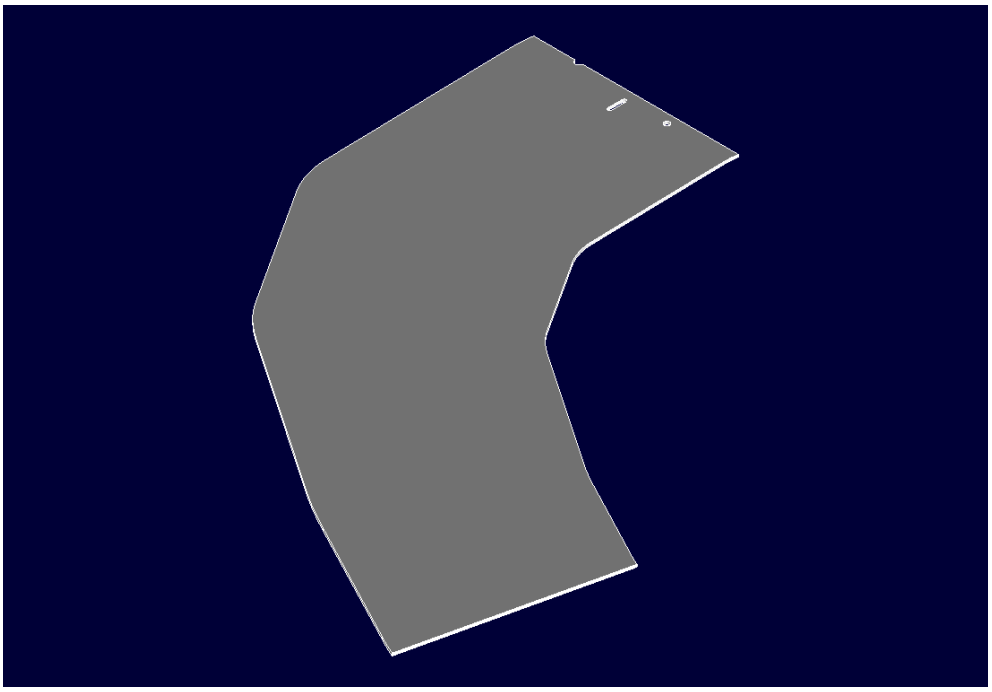


Figure 119 Machined flat with hole, crossing slot and edge notch

Ordinary machining on the developed blank

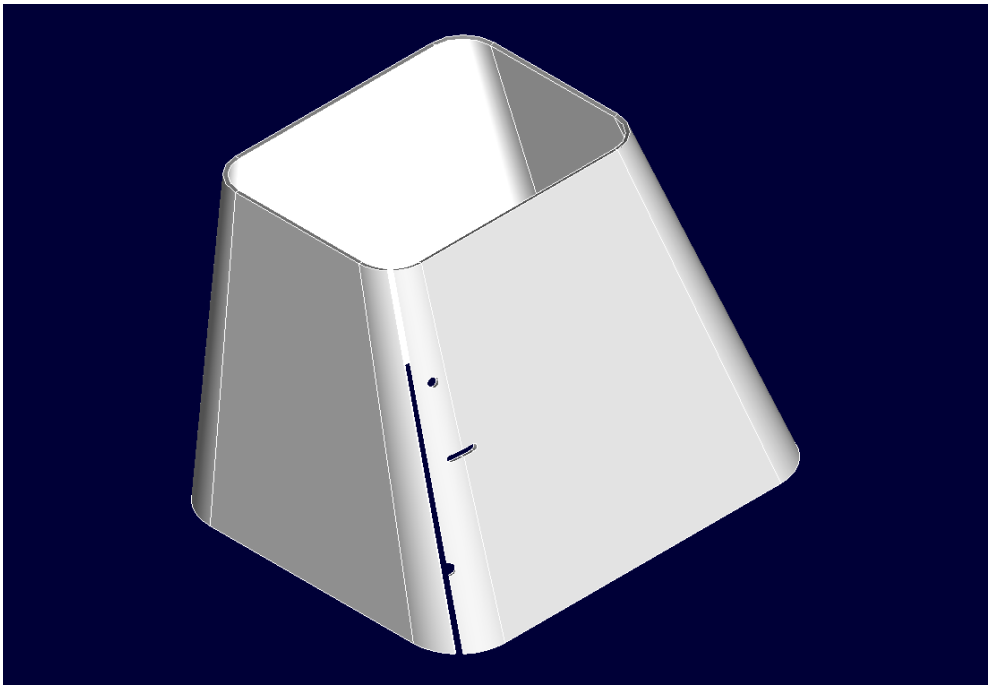


Figure 120 Refolded rectangle transition retaining machining

Machining retained after refolding

Section types and dimensions

Mode	Profiles / datum
Smooth cylinder / cone	Existing analytic coaxial roll. Inner radii and inner axial height; winding angle below 360°.
Round → round	Lower / upper neutral radii. Concentric or eccentric.
Rectangle → round	Lower neutral width / depth / corner R and upper neutral radius. Smooth or faceted.
Rectangle → rectangle	Lower / upper neutral width / depth / corner R. Smooth or faceted.

Create

• Select Create and the section type.

• Pick the lower section center. Profiles lie parallel to world XY; height is in +Z. Ordinary rigid move, rotate and mirror are available after creation.

• Enter section sizes and height. Upper center offsets X/Y are relative to the lower center and may be negative.

• Set the neutral seam gap. For triangulation, set 8–128 segments in multiples of 8. Rectangle corners are included; every interval has two triangles.

• Set thickness and K factor, then Apply. No geometry is calculated during parameter entry.

Loft dimensions specify the neutral surface. Inner and outer skins are offset normally by −K×t and +(1−K)×t. They are not inner or outer dimensions. Intersections of offset triangle planes form the joints; thickened end contours need not coincide with the neutral profiles.

Seam and accuracy

With Automatic seam position, for triangulation without corner R, the seam is centered on the +X side of each profile. Its gap is the straight distance between neutral endpoints, independently at the lower and upper ends. Rounded rectangle-to-round and rectangle-to-rectangle use the corner seam and smaller-radius gap convention described above. It must be smaller than the first segment interval. Specified seam position uses a direction angle or a point on the lower profile. Multiple seams and twisted profiles are not supported.

Circular profiles use straight chords. Maximum radial chord error is R×(1−cos(π/N)), displayed under Triangulation accuracy. At R40 mm it is about 0.193 mm for N32 and 0.0482 mm for N64. This measures circular-profile approximation, not overall smooth-surface or manufacturing accuracy.

Edit dimensions and corner radii after machining

• Choose Roll / loft → Edit existing roll and select the machined body, either unfolded or refolded.

• Change section dimensions, height, offsets or corner radii, then Apply. Rectangle-to-rectangle supports independent lower and upper radii.

• Unfold to inspect the machining. Undo restores both dimensions and machining.

Machining stays fixed in the loft's local developed XY coordinates.Hole, slot and notch cutting contours are neither moved, rotated nor scaled with the section dimensions. They do not follow a panel proportionally or maintain an automatic distance from a moving edge. Whole-part rigid moves, rotations and mirrors carry this coordinate frame.

Closed hole and slot contours retain their physical size. Moving the outside boundary can change how it meets a fixed notch; an expanded blank may surround it and turn it into a hole. Only the material actually removed by the original notch is known; no unseen extension of the extrusion tool is inferred. Stored footprints survive successive parameter changes, including returning to the original dimensions. Further ordinary machining recaptures footprints from the current actual material.

If a closed hole crosses the new boundary, a cut leaves the blank completely, or material becomes disconnected, Apply fails and retains the original body. Adjust machining in the flat when necessary. Newly saved machining metadata requires this version or later.

Unfold, refold and edit

Use Sheet metal unfold / refold. Faceted flats retain triangle boundaries; smooth flats have no internal triangulation guides. Use Roll / loft → Edit existing roll to load all dimensions, offsets, gap and segment count. Dimension editing works in folded and unfolded states. Use sheet metal settings to change thickness.

Through holes, slots and edge notches made while unfolded are retained by refold/unfold. Blind pockets, nonuniform thickness machining, material added outside the original blank and machining while folded are rejected. Loft dimension and radius edits retain machining in the developed coordinates described below. Finite-radius bend editing, edge flanges, forming features and partial unfolding are also outside this loft model. Loft dimension editing is available before and after flat machining. Whole-part rigid transformations remain available after machining. New loft JDC files require this version or later.

Drawings and DXF / DWG

Layer	Geometry
SM_CUT	Developed material perimeter.
SM_LOFT_GUIDE	Internal triangulation guides, including diagonals.
SM_LOFT_NOTE	Segment count, chord error and NO BRAKE ALLOWANCE note.

The Bend lines switch controls guides; Annotations controls notes. Guides are not cutting paths or finite-radius brake bends. Both DXF and DWG preserve separate layers. No B numbers or bend table are generated. Minimum bend-radius checking is unavailable for triangulated lofts.

Examples

Use thickness 1 mm, K0.4, N32, height 80 mm, offsets X20/Y10 mm and gap 2 mm. Round–round: lower R40 and upper R25; rectangle–round: lower 90×65 and upper R25; rectangle–rectangle: lower 90×65 and upper 50×40. Excessive offset/thickness, very short height or overlapping developments can be rejected.

Official references

• Fusion: lofted flange, die forming and brake forming

• Inventor: lofted flange types

Part 8 Customization guide

This chapter explains how to add drawing and reporting commands. Free executes only the release sample. Custom or modified add-ins require STD/Pro. MCP connection is available in Free.

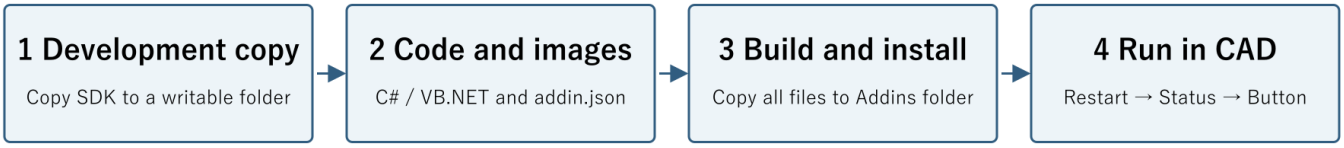


Figure 121 From source code to a running add-in.



Figure 122 The ten supplied API sample commands.



Figure 123 Check Add-in status and open the installation location with Add-in folder.

Choose an extension method

An add-in DLL runs inside JDesignCAD and registers ribbon buttons. A separate .NET application uses LocalCadClient. An AI client uses the supplied MCP executable. All three use the same public drawing API; the SDK does not contain a standalone CAD engine.

Free runs the ten built-in API sample commands. Custom DLLs and modified samples require an active STD or Pro license. In Free, replacing the sample DLL or copying its ID does not execute external code: JDesignCAD uses its embedded release sample. MCP remains available in Free, while licensed sheet-metal and analysis APIs are denied.

Install the supplied API sample

• Save your drawings and close JDesignCAD.

• Open the installed Customization folder, normally C:\Program Files\JDesignCADV1\Customization.

• Copy the complete SampleAddin folder into %LOCALAPPDATA%\JDesignCAD\Addins\JDesignCAD.Customization.Sample. Alternatively run Install-Sample.ps1 from PowerShell.

• Restart CAD. Open Customization → Add-in status, then check the API sample tab.

• Open a drawing and select the target entities before running a sample command.

```
powershell -File "C:\Program Files\JDesignCADV1\Customization\Install-Sample.ps1"
```

The script stops if the destination already exists. Preserve any edited source or existing add-in before updating it manually. The sample tab and its commands currently use Japanese labels in both UI languages.

Prepare Visual Studio

• Install Visual Studio 2022 version 17.8 or newer with the .NET desktop development workload and .NET 8 SDK.

• Copy Customization\SDK to a writable development directory. Keep Api and Samples in their original relative positions.

• Open Samples\Customization.Workshop\Customization.Workshop.csproj for C#, or Samples\Customization.WorkshopVB\Customization.WorkshopVB.vbproj for VB.NET.

• Build Release for x64. The practical ten-command sample is Samples\Customization.Sample\Customization.Sample.csproj.

Reference SDK\Api\JDesignCAD.Api.dll with Copy Local=False. Keep the XML file beside the DLL for IntelliSense. Ribbon registration requires no reference to internal JDesignCore or Krypton types. Class libraries cannot be run directly with F5.

Create a new project

Select a .NET class library, not a .NET Framework class library. Set the target framework and reference as below. Add UseWindowsForms=true when you create Windows Forms input dialogs.

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net8.0-windows</TargetFramework>
    <PlatformTarget>x64</PlatformTarget>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <EnableDynamicLoading>true</EnableDynamicLoading>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="JDesignCAD.Api">
      <HintPath>..\..\Api\JDesignCAD.Api.dll</HintPath>
      <Private>false</Private>
    </Reference>
    <None Update="addin.json;*.png" CopyToOutputDirectory="PreserveNewest" />
  </ItemGroup>
</Project>
```

Register a tab and button

Implement ICadAddin with a public parameterless constructor. OnStartup registers a RibbonTab containing RibbonPanel groups and RibbonButton commands. Each command implements IAddinCommand.ExecuteAsync. Keep startup registration short; perform drawing work when the user presses the button.

Use await in C# or Await in VB.NET. Do not block the UI thread with Task.Result or Wait(). The API dispatches drawing changes to the CAD UI thread. Use StartOperationAsync for heavy geometry calculations and commit a ready result explicitly.

Configure addin.json

```
{
  "id": "MyCompany.MyTool",
  "name": "My drawing tools",
  "apiVersion": 1,
  "assembly": "MyTool.dll",
  "entryType": "MyCompany.MyTool.MyAddin",
  "enabled": true
}
```

Field	Meaning
id	Unique add-in identifier. Retain it for updates to the same add-in; use your own ID when editing a sample.
apiVersion	Major version 1, including API 1.2 and 1.3.
assembly	DLL path relative to this add-in folder.
entryType	Fully qualified public ICadAddin implementation with a parameterless constructor.
enabled	false disables startup loading.

VB.NET combines RootNamespace with the class namespace. The supplied VB sample omits a Namespace block to avoid duplicating the root namespace.

Set button images

RibbonButton.SmallIconPath and LargeIconPath are paths relative to the add-in directory. Supply 24×24 and 48×48 PNG images for the usual small/large sizes. Omitted paths leave the button without an image. The first button image is also used when its group collapses.

```
var button = new RibbonButton("line", "Line", "Draw a line", command,
    SmallIconPath: "line24.png", LargeIconPath: "line48.png");
```

Place the PNG files beside addin.json or in its subfolder. Paths cannot escape the add-in directory. These are runtime images from the add-in folder; they are not entries in the application's Resources.resx.

Build install and debug

• Build Release/x64, then save drawings and close CAD.

• Create %LOCALAPPDATA%\JDesignCAD\Addins\MyCompany.MyTool.

• Copy your DLL, addin.json, generated deps.json, required dependency DLLs and icons. Share the host's JDesignCAD.Api.dll.

• Restart CAD and check Customization → Add-in status. Use an active STD/Pro license for a custom or modified DLL.

• For debugging, deploy the Debug DLL and PDB, then attach Visual Studio to JDesignCAD.exe as managed .NET code. Check the loaded module path and symbols if a breakpoint is not hit.

Updates require restarting CAD. The loader isolates dependency contexts; this is not a security sandbox. Add-ins run with the user's Windows privileges. The startup log is %LOCALAPPDATA%\JDesignCAD\Automation\addins.log.

Ten practical sample commands

Ribbon label	Behavior and scope
文字連番	Number selected ordinary text while retaining positions and formatting.
文字置換	Case-sensitive replacement in selected text.
大文字・小文字	Change selected Latin text to upper or lower case.
文字囲み	Frame WCS XY text with rectangles or circles.
等間隔複写	Copy by WCS X/Y/Z spacing and count; retain originals.
同種・同層選択	Select the same entity type and layer as one reference entity.
重複線整理	Review and remove duplicate lines/circles with matching attributes; partial overlaps are excluded.
色を ByLayer	Set only selected entity colors to ByLayer.
長さ・面積集計	Calculate curve lengths and closed profile areas in a worker.
ブロック集計	Count named block references directly in model space, not recursive nested occurrences.

Revision and retry rules

GetDocumentAsync returns a session-scoped ID and Revision. Use the current values to construct a request and a new UUID for each new edit. After a transport failure, resend the exact same request and operationId to recover its result. Do not generate a new UUID blindly. A different request with the same operationId is rejected. Requery face/edge references after any revision change.

Coordinate values are WCS millimetres even when the UI uses another unit. Angles are degrees. Successful edits are atomic Undo batches. Undo includes manual user actions. Read-only queries and selection changes do not add an Undo step.

Shared errors and recovery

Code	What to do
no_document / unsupported_view	Open the intended drawing in model space.
cad_busy	Finish or cancel the current command/dialog.
document_changed / revision_conflict	Reinspect the drawing and rebuild the request from current IDs.
entity_not_found / entity_not_in_model	Requery top-level entity IDs in this document session.
layer_locked / undo_disabled	Review drawing state before editing.
license_required	Read LicenseRestriction and RequiredPlans. Do not retry unchanged requests to bypass the restriction.
geometry_failed / invalid_result	Review dimensions, topology and engine-supported geometry. The calculation does not guarantee success for every BRep input.
worker_unavailable	Use a complete installation containing matching Automation.Worker files.
message_too_large	Reduce page or request size; the local bridge limit is 1 MiB per message.

C# complete sample

This is source from the supplied SDK. Use its matching project and addin.json; an active STD/Pro license is required to load the built DLL. Long lines wrap for printing.

```
using JDesignCAD.Api;
namespace Customization.Workshop;

public sealed class WorkshopAddin : ICadAddin
{
    public void OnStartup(IAddinContext context)
    {
        context.RegisterTab(new("workshop","API 実習",
        [new("create","作成",[
            new("curves","線と円","1 回の Undo で取り消せる線と円",new WorkshopCommand("curves")),
            new("box","箱を作成","別プロセスで 20×30×10 mm の箱を計算",new WorkshopCommand("box")),
            new("loft","角–角ロフト","STD/Pro。滑らかな角–角ロフトを作成",new WorkshopCommand("loft"))]),
         new("inspect","確認",[
            new("selection","選択情報","選択数と図面のリビジョン",new WorkshopCommand("selection")),
            new("undo","元に戻す","直前の操作を取り消し（手動操作も対象）",new WorkshopCommand("undo"))]) ]));
    }
    public void OnShutdown() { }
}
public sealed class WorkshopCommand(string action) : IAddinCommand
{
    public async Task<CommandResult> ExecuteAsync(ICadApi api,CancellationToken cancellationToken)
    {
        if(action=="selection")
        {
            var selected=await api.GetSelectionAsync(cancellationToken:cancellationToken);
            return new($"選択 {selected.Total} 個 / revision {selected.Document.Revision}");
        }
        var document=await api.GetDocumentAsync(cancellationToken);
        if(action=="undo")
        {
            await api.UndoAsync(new(document.Id,document.Revision,Guid.NewGuid().ToString("D")),cancellationToken);
            return new("直前の操作を取り消しました。");
        }
        if(api is not ICadJobApi automation)throw new InvalidOperationException("API v1.2 対応の JDesignCAD が必要です。
");
        if(action=="curves")
        {
            var request=document.Operation("geometry.create",new CreateObjectsParameters([
                new(Line:new(new(0,0,0),new(50,0,0))),new(Circle:new(new(25,20,0),10,new(0,0,1))) ]));
            await automation.ExecuteOperationAsync(request,cancellationToken);
            return new("線分と円を作成しました。");
        }
        var operation=action=="box"?document.Operation("solid.primitive",new
PrimitiveParameters("box",new(0,0,0),new(20,30,10)))
            :document.Operation("sheet.loft",new
SheetLoftParameters(new(),new(Kind:3,FormingMode:1,Height:80,BottomWidth:100,BottomDepth:80,TopWidth:60,TopDepth:40,Co
rnerRadius:10,TopCornerRadius:6)));
        var capability=(await
automation.GetCapabilitiesAsync(cancellationToken)).Operations.Single(o=>o.Name==operation.Operation);
        if(!capability.Available)throw new InvalidOperationException(capability.UnavailableReason);
        var job=await automation.StartOperationAsync(new(operation,120),cancellationToken);
        try
        {
            while(job.State=="running")
            {
                await Task.Delay(250,cancellationToken);
                job=await automation.GetJobAsync(job.Id,cancellationToken);
            }
            if(job.State!="ready")throw new InvalidOperationException(job.ErrorCode+": "+job.Message);
            // Refuse to apply a result to a drawing changed during calculation.
```

```
            var current=await api.GetDocumentAsync(cancellationToken);
            await
automation.CommitJobAsync(new(current.Id,current.Revision,Guid.NewGuid().ToString("D"),job.Id),cancellationToken);
        }
        catch
        {
            try { await automation.CancelJobAsync(job.Id,CancellationToken.None); } catch { /* CAD may be closing.
*/ }
            throw;
        }
        return new("作成しました。1 回の［元に戻す］で取り消せます。");
    }
}
```

VB.NET complete sample

This is source from the supplied SDK. Use its matching project and addin.json; an active STD/Pro license is required to load the built DLL. Long lines wrap for printing.

```
Imports System.Threading
Imports System.Threading.Tasks
Imports JDesignCAD.Api

Public NotInheritable Class WorkshopAddin
    Implements ICadAddin
    Public Sub OnStartup(context As IAddinContext) Implements ICadAddin.OnStartup
        Dim button = New RibbonButton("line", "線分作成", "50 mm の線分を作成", New LineCommand())
        Dim panel = New RibbonPanel("basic", "VB.NET", New RibbonButton() {button})
        context.RegisterTab(New RibbonTab("workshop-vb", "API 実習 VB", New RibbonPanel() {panel}))
    End Sub
    Public Sub OnShutdown() Implements ICadAddin.OnShutdown
    End Sub
End Class

Public NotInheritable Class LineCommand
    Implements IAddinCommand
    Public Async Function ExecuteAsync(api As ICadApi, cancellationToken As CancellationToken) As Task(Of
CommandResult) Implements IAddinCommand.ExecuteAsync
        Dim doc = Await api.GetDocumentAsync(cancellationToken)
        Dim request = New CreateGeometryRequest(doc.Id, doc.Revision, Guid.NewGuid().ToString("D"),
            New LineDefinition() {New LineDefinition(New Point3(0, 0, 0), New Point3(50, 0, 0))},
            Array.Empty(Of CircleDefinition)())
        Await api.CreateGeometryAsync(request, cancellationToken)
        Return New CommandResult("線分を作成しました。元に戻すで取り消せます。")
    End Function
End Class
```

Part 9 MCP integration guide

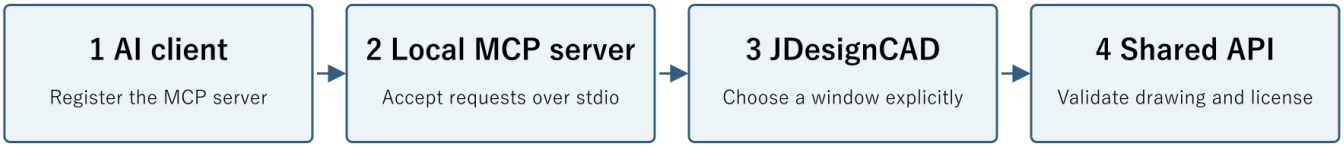


Figure 130 How an AI request reaches CAD.

File input and output: save/export, five methods and MCP tools, formats, completion and errors

What MCP connects

The supplied JDesignCAD.Mcp.exe exposes 31 tools to an MCP-capable AI client. The client starts it over stdio. The server connects to a running CAD window through a same-Windows-user named pipe; it opens no TCP listening port. It does not provide a CAD engine or open a target drawing automatically.

Free can use MCP. Sheet-metal and analysis operations require active STD/Pro permissions, including when called through MCP. The Free restriction on loading custom ribbon DLLs does not disable the MCP server.

Register the server

• Install the complete JDesignCAD 1.1.4 package. Check that Customization\Mcp\JDesignCAD.Mcp.exe and its accompanying files exist.

• Open your AI client's MCP server configuration. Use the executable path in the example, changing it if CAD was installed elsewhere.

• Restart or reconnect the MCP client as required by that client.

• Start JDesignCAD as the same Windows user and open a test drawing in model space.

• Ask the client to list CAD sessions and select the intended window explicitly.

```
{
  "mcpServers": {
    "jdesigncad": {
      "command": "C:\\Program Files\\JDesignCADV1\\Customization\\Mcp\\JDesignCAD.Mcp.exe",
      "args": []
    }
  }
}
```

The supplied Customization\mcp-config.json is a configuration example. The surrounding configuration file and setup UI depend on the MCP client. Keep the complete Mcp directory together. Do not run only the executable copied away from its DLLs. No special network endpoint or API token is required by this local bridge.

Inspect before editing

• cad_list_sessions returns candidate windows. Choose sessionId explicitly; do not assume the first result is correct.

• cad_get_document returns documentId, revision, object count and Undo/Redo state.

• cad_get_capabilities returns registered operation names, execution modes and license availability.

• cad_get_operation_schema returns exact JSON parameters for an operation.

• For an edit, create a new UUID operationId and use the latest document ID and expectedRevision.

Coordinates and lengths are WCS millimetres regardless of the displayed drawing unit. Angles are degrees. IDs are session-scoped and face/edge references expire when the revision changes.

Create one line

Call cad_execute_operation with the following structure. Replace the marked IDs, revision and UUID with live values. The line is created in one Undo batch.

```
{
  "sessionId": "TARGET_SESSION",
```

```
  "request": {
    "documentId": "CURRENT_DOCUMENT",
    "expectedRevision": 0,
    "operationId": "NEW_UUID",
    "operation": "geometry.create",
    "parameters": {
      "objects": [{"line": {
        "start": {"x": 0, "y": 0, "z": 0},
        "end": {"x": 50, "y": 0, "z": 0}
      }}]
    },
    "selectResult": true
  }
}
```

Check the returned EditResult and updated document state. Do not interpret a dispatched GUI command as proof that a line was created. An example AI instruction is: “In the selected test drawing, create one 50 mm line along WCS X from the origin. Confirm the drawing before editing and report the returned entity ID.”

Calculate and then commit

Heavy operations use cad_start_operation with a request and timeoutSeconds, normally 120. Poll cad_get_job at a reasonable interval. running means queued or calculating. ready means that geometry is available but the drawing is unchanged. Verify that the drawing revision still matches the source, then call cad_commit_job with a new commit UUID. The start UUID must not be reused for the commit.

State	Next action
running	Wait and query again, or cancel.
ready	Commit explicitly or discard with cad_cancel_job.
committed	The drawing edit was applied. Repeating the identical commit request does not duplicate it.
completed	A read-only data result is available; read cad_get_job_data. Do not commit.
failed / cancelled	Inspect errorCode/message; the calculation did not modify the drawing.

At most four running/ready jobs are retained; one worker calculation runs at a time. Timeouts range from 1 to 1800 seconds. Start requests capture source geometry and revision. Merely supplying a newer revision to an old result cannot make it valid.

License errors visible to the AI

cad_get_capabilities includes license.currentPlan and operation availability. A denied operation has unavailableCode=license_required and licenseRestriction with the feature, operation, current plan and required plans. These details are also returned by schema and interactive-command discovery. Execution and committing a ready result recheck permissions.

```
{
  "code": "license_required",
  "message": "This API requires an active STD or Pro license.",
  "licenseRestriction": {
    "feature": "sheetmetal",
    "operation": "sheet.loft",
    "currentPlan": "Free",
    "requiredPlans": ["STD", "Pro"],
    "retryable": false
  }
}
```

This is an illustrative payload. MCP returns isError=true and a JSON error text. The AI should explain the restriction and required plan, then offer an available operation. Repeating requests or trying another API does not remove the restriction. The rejected operation leaves the drawing unchanged.

Transport failure and Undo

When a connection drops during an edit, resend exactly the same request and operationId to retrieve its result. Generating a new UUID blindly can duplicate edits. Worker calculations do not automatically stop when the client disconnects; retry the identical start request to recover the job ID.

cad_undo and cad_redo operate on the drawing's global history, including manual edits. They are not an MCP-only history. Query the current document state before using them, and do not undo an unrelated user action.

GUI commands are interactive

cad_get_interactive_commands lists GUI command IDs; cad_launch_interactive_command queues one for the user to complete. The result queued is an acknowledgement only. It does not confirm saving, printing, creation or a successful solve. Prefer structured operations from cad_get_capabilities for automation.

Troubleshooting

Symptom	Check
No server tools	Executable path, complete Mcp folder, desktop runtime and client configuration.
No CAD sessions	CAD is running under the same Windows user. Open a drawing and reconnect the client.
cad_busy	Finish or cancel the active CAD command or dialog.
session_not_found / cad_unavailable	List sessions again; do not silently switch to a different drawing.
revision_conflict	Reinspect entities/topology and restart from the current drawing state.
license_required	Explain requiredPlans and select an available operation.
message_too_large	Reduce request/page size. Each bridge message is limited to 1 MiB.
worker_unavailable	Restore the matching Automation.Worker files from the complete installer.
geometry_failed / invalid_result	Review geometry, dimensions and operation limits. No universal BRep success guarantee is implied.

Entity list pages contain up to 1000 items; topology and job-data pages up to 256. Large analysis inputs or transient frames can exceed the bridge limit. Unknown operations and arbitrary calls to internal CAD methods are not accepted.

Save and export

cad_get_file_formats → cad_save_document or cad_export_document → cad_get_file_operation. Cancel with cad_cancel_file_operation. Only completed confirms success. Inspect error for failed operations; do not use cad_commit_job for file output. Free formats are JDC, JSON and STL.

Inspect drawings through MCP

Execute view.capture with cad_execute_operation to render the model or a named layout to a local PNG. Layout images include native views, paper geometry and dimensions. Geometry, selection and revision are unchanged. Supply an absolute filePath, optional layoutName, width (320–4096), height (240–4096), and overwrite only when replacing an existing file. The folder must exist. Open the resulting image in the client to inspect it.

Use cad_query with kind layouts for available sheets or sheetDefaults for current sheet-metal settings. layout.create creates native model views, paper objects and scaled dimensions as one undoable operation. layout.activate switches sheets. Partial sheet.flange ranges use startDistance and endDistance. Discover each operation schema before execution.

Drafting rules for notes and dimensions

Use native text leaders for feature notes such as radii, hole counts, slots and chamfers. Query layoutAnnotations for current paper indices, then call layout.annotations at the same revision. Each leader has Text, Anchor (arrow tip), Landing (start of text shelf) and optional Elbows. Text, arrows and segments remain one editable annotation. Existing dimensions, models and viewports are preserved.

For circle/arc location or pitch dimensions, use exact centers when appropriate to the intended dimension, not tangent endpoints with coincident projected coordinates. Follow explicit overall, tangent or edge dimensions in the source drawing. Preserve existing dimensions when requested by the user. Inspect the result using view.capture.

Open drawings and switch model/layout views

Discover these operations with cad_get_capabilities and cad_get_operation_schema, then use the existing cad_execute_operation tool. No additional MCP connection configuration is required.

• cad_query, kind="views": lists names, model/layout kinds, active status and paper dimensions.

• view.activate: {"kind":"model"} returns to the model; {"kind":"layout","name":"Sheet name"} displays a layout. Camera and zoom are preserved unless zoomToFit=true is requested. Geometry and Undo history are unchanged.

• document.open: {"filePath":"D:/Drawings/drawing.jdc"} opens an existing native JDC or JDC JSON in the same window. Detached geometry loads in the background before installation. Opening is not an undoable edit.

Supply the current documentId and expectedRevision and a fresh operationId. Only when no drawing is open on Home, use an empty documentId and expectedRevision=0. Unsaved edits cause unsaved_changes: first save with cad_save_document and wait for completed. Edits or document switches during loading also prevent replacement. After a transport timeout, retry the identical request and operationId; the drawing will not open twice. Use the new document identity in the successful response and reacquire entity IDs.

Apply text or images to faces

Select Surface label from the annotation button dropdown, click a model face, choose text or an image, and preview its width, height, rotation and center before completing the command. Choose an existing label in Target label, select Edit or Remove, and commit with the checkmark. Face selection and settings changes do not commit automatically. The font dropdown initially uses the current drawing text font. Removing a label preserves its owner geometry and is undoable. The assignable command ID is MenuId.Constr.SurfaceLabel.

Planes, cylinders, spheres and freeform surfaces are supported. Labels continue across connected patches of the same surface and cylindrical seams, while real boundaries and holes clip the image. Dimensions use the tangent metric at the center; spherical/freeform labels may distort away from it. Singular centers and placements spanning a full period are rejected.

Labels do not change BRep geometry or volume. Embedded pixels survive JDC save/reload without the source image, follow move/copy, and appear in printing and PNG output. STEP/STL do not include raster labels.

Use cad_execute_operation with face.label. Obtain a current face reference from topology (entityId, revision, kind="face", index). Supply WCS center, width/height in mm, rotationDegrees, and either text or an absolute imagePath. Text also supports fontName and a six-digit RGB color. Discover the full schema first.

Query kind="surfaceLabels" with 1–100 explicit entityIds for face references, label IDs and placement. Mode add is the default; replace/remove require a labelId. Changes across connected patches form one Undo step.

MCP tool reference

The 31 tools use camelCase JSON. Specify the target CAD with sessionId and check the document revision and operation definition before editing.

cad_cancel_file_operation

Cancel a running file save/export before destination commit. Stops its worker; the previous destination remains. completed operations cannot be undone by this tool. Query its terminal state after cancellation.

Input	Type	Required / default
sessionId	string	Required
operationId	string	Required

Corresponding method: CancelFileOperationAsync

cad_cancel_job

Cancel an owned computation or discard its uncommitted ready result. Leaves drawing geometry unchanged.

Input	Type	Required / default
sessionId	string	Required
jobId	string	Required

Corresponding method: CancelJobAsync

cad_commit_job

Apply a ready worker result as ONE Undo step. Fails if drawing changed since job start. Consumes only explicitly documented source objects. Requires a fresh commit operationId; identical retries do not duplicate results.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: CommitJobAsync

cad_create_curves

Create 3D linear polylines and circular arcs as ONE Undo batch. WCS mm. Closed polyline: >=3 vertices, do not repeat first at end. Arc: center/start/normal and right-hand signed sweepDegrees with 0<abs(sweep)<360; start-center perpendicular to normal. 1..1000 curves and <=4096 total polyline vertices. Empty arrays for unused types. Requires fresh document revision and UUID operationId; identical retries only.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: CreateCurvesAsync

cad_create_geometry

Create lines and/or circles in model space as ONE undoable atomic batch. All coordinates/radii are WCS millimetres. A circle has center, radius and nonzero plane normal (use {x:0,y:0,z:1} for XY). Supply documentId/expectedRevision from a fresh document query and a new UUID operationId. Use empty arrays for unused types. Max 1000 objects. On connection loss retry the EXACT request and operationId; do not blindly repeat with a new ID.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: CreateGeometryAsync

cad_delete_entities

Delete 1..1000 explicit top-level IDs as ONE Undo batch. Query/inspect intended objects first. Rejects stale revisions, removed IDs and locked layers. Requires documentId, expectedRevision and UUID operationId; use identical retries after connection loss. Returned entity metadata describes removed objects; Undo restores them.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: DeleteEntitiesAsync

cad_execute_operation

Execute a registered immediate operation as one Undo step. Obtain operation names from cad_get_capabilities and use their documented parameter records (camelCase). Requires current document ID/revision and a fresh operationId UUID. Retry identical request after connection loss. Unknown fields and unregistered operations are rejected.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: ExecuteOperationAsync

cad_export_document

Start export to an absolute filePath with explicit format and overwrite (default false). Does not change drawing filename/dirty state/Undo. Null entityIds exports whole project; explicit IDs export a selection with optional WCS-mm basePoint. STEP/mesh/SVG/IFC use active model; IFC is ONE component. Check cad_get_file_formats first. Poll cad_get_file_operation; running is NOT completion. Same operationId only for identical retries. Errors preserve destination; licensed formats require STD/Pro.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: ExportDocumentAsync

cad_get_attributes

Read color mode, line style/width, layer and lock state for explicit top-level entity IDs.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: GetAttributesAsync

cad_get_capabilities

List operations implemented by this CAD host, required parameter record, execution mode, and availability. Check before issuing an operation; do not guess operation names. Modeling rule: when one physical part is built from multiple solids, you must unite its connected pieces with solid.boolean (kind=union) once that part's machining and any downstream use of the separate pieces are complete, during construction rather than only after the whole assembly. Verify a successful single-solid result and refresh entity IDs/revision after commit. Keep distinct physical parts separate, including separate instances of identical parts, nuts, washers and moving parts. Honor an explicit request to keep pieces separate. Do not add bridges or alter dimensions to join disconnected pieces without user authorization. If union fails, preserve the inputs and report the unresolved union; a block/group or overlapping solids is not a completed Boolean union.

Input	Type	Required / default
sessionId	string	Required

Corresponding method: GetCapabilitiesAsync

cad_get_document

Get the target drawing's document ID, current revision, object count and Undo/Redo state. Coordinates/lengths in this API are WCS millimetres regardless of display units.

Input	Type	Required / default
sessionId	string	Required

Corresponding method: GetDocumentAsync

cad_get_entities

List top-level objects in drawing order with IDs, types, layers and WCS mm bounds. Supply documentId/expectedRevision from cad_get_document. Limit 1..1000; use the same revision for all pages to prevent mixing drawing states.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: GetEntitiesAsync

cad_get_entity_details

Get exact line endpoints, circle center/radius/normal, signed arc sweep in degrees, or linear polyline vertices/closure for 1..100 top-level IDs. Coordinates are WCS mm. geometryStatus=unsupported for other types, too_large if the 4096-point response budget is exceeded; metadata remains available. IDs come from selection/entities/create in this document session.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: GetEntityDetailsAsync

cad_get_file_formats

Discover automated save/export formats and current license restrictions. Save changes the current filename only when completed; export writes a copy. Do not guess formats. Use cad_save_document/cad_export_document then poll cad_get_file_operation.

Input	Type	Required / default
sessionId	string	Required

Corresponding method: GetFileFormatsAsync

cad_get_file_operation

Poll file output by operationId in the originating CAD session. Only state=completed confirms success (filePath, bytes, sha256, warnings). failed/cancelled includes error.code/message/licenseRestriction and MUST be reported to the user as unsuccessful. No geometry-job commit is needed; do not call cad_commit_job. A completed result is historical and is never a request to rewrite the file.

Input	Type	Required / default
sessionId	string	Required
operationId	string	Required

Corresponding method: GetFileOperationAsync

cad_get_interactive_commands

List registered CAD GUI command IDs and availability. These launch interactive tools/dialogs; they are NOT automated geometry APIs. Prefer cad_get_capabilities for headless parameterized operations.

Input	Type	Required / default
sessionId	string	Required

Corresponding method: GetInteractiveCommandsAsync

cad_get_job

Get worker state: running, ready, failed, cancelled, committed, or completed (read-only data). A ready result has not changed the drawing yet; use cad_get_job_data for completed data jobs.

Input	Type	Required / default
sessionId	string	Required
jobId	string	Required

Corresponding method: GetJobAsync

cad_get_job_data

Read completed worker-data job: section summary or items, limit 1..256. Data jobs do not modify the drawing or require commit; results describe the sourceRevision captured at start.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: GetJobDataAsync

cad_get_layers

List drawing layers and available line pattern/width names. Requires current document ID and revision.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: GetLayersAsync

cad_get_operation_schema

Get JSON Schema for a registered operation's parameters. Use this after capability discovery and before constructing cad_execute_operation parameters. Parameter names are camelCase; only listed fields are accepted.

Input	Type	Required / default
sessionId	string	Required
operation	string	Required

Corresponding method: GetOperationSchemaAsync

cad_get_selection

Get selected objects as IDs, types, layers and WCS bounding boxes in millimetres. IDs are scoped to the open drawing session. Finish any active CAD command first. Paging limit: 1..1000.

Input	Type	Required / default
sessionId	string	Required
offset	integer	0
limit	integer	256

Corresponding method: GetSelectionAsync

cad_get_topology

List faces/edges/vertices of a top-level solid/shell/face. References expire when the document revision changes. Explicit kind and paging; limit 256.

Input	Type	Required / default
sessionId	string	Required

Input	Type	Required / default
request	object	Required

Corresponding method: GetTopologyAsync

cad_launch_interactive_command

Launch a registered CAD GUI command for the user to complete manually. May open dialogs or begin selection. queued is NOT success/completion of drawing/saving/printing. Prefer automated operations when available. Requires current document/revision and unique operationId; retry identical request only.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: LaunchInteractiveCommandAsync

cad_list_sessions

List running JDesignCAD windows for the current Windows user. Choose an explicit sessionId before accessing a drawing.

Corresponding method: LocalProtocol.FindSessions

cad_query

Inspect exact geometry (including spline poles/knots), named blocks, sheet-metal rules or styles. Kind geometry/sheet: explicit 1..100 IDs. Kind blocks/styles: empty IDs. WCS mm. Query statuses distinguish unsupported and too_large.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: QueryAsync

cad_redo

Redo the latest undone operation in the target drawing. Requires current documentId and expectedRevision plus a fresh UUID operationId. Retry identical requests after connection loss.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: RedoAsync

cad_save_document

Start saving the COMPLETE drawing to an absolute filePath with explicit format and overwrite (default false). Requires fresh document/revision and UUID operationId. Returns running, not saved: poll cad_get_file_operation until completed/failed/cancelled. Only completed changes the current filename and clears dirty state. Retry identical requests with the SAME operationId; never blindly submit another ID. No dialogs; Free restrictions apply.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: SaveDocumentAsync

cad_set_selection

Replace the current CAD selection with explicit top-level entity IDs, or [] to clear. Changes highlight only; no Undo step. Requires documentId, expectedRevision and a new UUID operationId; retry the identical request after connection loss.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: SetSelectionAsync

cad_start_operation

Start an isolated worker calculation on detached geometry. Does not change the drawing. Discover worker operation and parameter schema first. Poll cad_get_job; commit a ready result explicitly. Requires source document/revision and unique operationId; retry identical requests. Timeout 1..1800 seconds. Modeling rule: when one physical part is built from multiple solids, you must unite its connected pieces with solid.boolean (kind=union) once that part's machining and any downstream use of the separate pieces are complete, during construction rather than only after the whole assembly. Verify a successful single-solid result and refresh entity IDs/revision after commit. Keep distinct physical parts separate, including separate instances of identical parts, nuts, washers and moving parts. Honor an explicit request to keep pieces separate. Do not add bridges or alter dimensions to join disconnected pieces without user authorization. If union fails, preserve the inputs and report the unresolved union; a block/group or overlapping solids is not a completed Boolean union.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: StartOperationAsync

cad_translate_entities

Move (copy=false) or copy (copy=true) 1..1000 explicit top-level objects by WCS mm displacement as ONE Undo batch. Preserves attributes; locked layers are rejected. Face/edge subobjects are not supported. Requires current documentId/expectedRevision and UUID operationId. Identical request retry prevents duplicate moves/copies. selectResult optionally selects results.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: TranslateEntitiesAsync

cad_undo

Undo the latest operation in the target drawing, including manual CAD operations. Requires current documentId and expectedRevision plus a fresh UUID operationId. This is not restricted to changes made by MCP. Retry identical requests after connection loss.

Input	Type	Required / default
sessionId	string	Required
request	object	Required

Corresponding method: UndoAsync

Part 10 API method and operation reference

Prepare to run the examples 472

Settings and connection 473

Drawing: lines, circles and closed profiles 478

Selection and geometry inspection 481

Input and interactive commands 488

Editing, copying and history 492

3D and background operations 496

Sheet metal 501

Measurement and analysis 502

Layouts, annotations and preview 504

Open, save and export 505

Command registration and add-ins 507

Method index 509

Browse settings, drawing, selection, input, editing, 3D, sheet metal, analysis, layouts and file output. Method details and working examples are grouped by purpose.

Prepare to run the examples

The supplied Samples.cs compiles against JDesignCAD.Api.dll on .NET 8 with implicit usings. It uses System, LINQ, threading/tasks, collections, System.Text.Json and JDesignCAD.Api. Connect externally with Connect(processId), or cast IAddinContext.Api to ICadFileApi inside an add-in. Pass CancellationToken.None or a token from your own CancellationTokenSource.

Run geometry-changing examples in a new practice drawing. Use a fresh OperationId UUID; only an identical transport retry reuses it. Never hide a conflict by changing only the revision. Inspect CadApiException.Code and distinguish availability, licensing and invalid geometry.

In ApiExamples, run dotnet build. Open a practice drawing in JDesignCAD and run dotnet run -- --list to find its ProcessId. For example, dotnet run -- 1234 drawing creates a line and circle; replace 1234 with the actual ProcessId. inspect reads the selection, rectangle draws a closed profile, box creates a solid and sheet creates a sheet-metal base. Program.cs is the entry point; Samples.cs contains the implementations.

Settings and connection

Pass a ProcessId obtained from FindSessions to Connect; do not silently choose the first window. Lengths are WCS millimetres regardless of display units, and angles are degrees. SettingsAsync creates or updates the Example layer; inspect Layers in the returned page.

Working example: connect

```
public static ICadFileApi Connect(int processId)
 {
     SessionInfo session = LocalProtocol.FindSessions()
         .Single(s => s.ProcessId == processId);
     return new LocalCadClient(session);
 }
```

Working example: settings

```
public static async Task<LayerPage> SettingsAsync(
     ICadAutomationApi cad, CancellationToken ct = default)
 {
     CapabilityInfo caps = await cad.GetCapabilitiesAsync(ct);
     if (!caps.Operations.Any(o => o.Name == "layers.set" && o.Available))
         throw new InvalidOperationException("layers.set is unavailable.");
     OperationSchema schema = await cad.GetOperationSchemaAsync("layers.set", ct);
     Console.WriteLine(schema.ParametersSchema.GetRawText());
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     EditResult edit = await cad.ExecuteOperationAsync(doc.Operation("layers.set",
         new SetLayerParameters("Example", new CadColor("rgb", "0080C0"))), ct);
     return await cad.GetLayersAsync(
         new DocumentRequest(edit.Document.Id, edit.Document.Revision), ct);
 }
```

ICadApi.GetDocumentAsync

Read the active drawing — Returns the document ID, revision, object count and Undo/Redo availability. Call immediately before constructing an edit request.

Declaration

```
Task<DocumentInfo> GetDocumentAsync(

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<DocumentInfo>. Await the Task.

Property	Type
Id	string
FileName	string
Revision	long
ObjectCount	int
CanUndo	bool
CanRedo	bool

Property	Type
LengthUnit	string
CoordinateSystem	string

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadAutomationApi.GetCapabilitiesAsync

Discover operations and license conditions — Returns registered names, parameter types, execution modes, availability, restriction details and current plan. Do not invent operation names; check Available.

Declaration

```
Task<CapabilityInfo> GetCapabilitiesAsync(

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<CapabilityInfo>. Await the Task.

Property	Type
Major	int
Minor	int
CoordinateSystem	string
LengthUnit	string
Operations	OperationInfo[]
Limitations	string[]
License	LicenseCapabilities

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadAutomationApi.GetOperationSchemaAsync

Read an operation schema — Returns JSON Schema for a registered operation, including required fields, types, nullability and nested objects. JSON names use camelCase.

Declaration

```
Task<OperationSchema> GetOperationSchemaAsync(

    string operation,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
operation	string	Registered operation name from GetCapabilitiesAsync.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<OperationSchema>. Await the Task.

Property	Type
Operation	OperationInfo
ParametersSchema	Text.Json.JsonElement

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadAutomationApi.GetLayersAsync

List layers and line styles — Returns layer definitions and available line pattern/width names. Use these names when changing drawing attributes.

Declaration

```
Task<LayerPage> GetLayersAsync(

    DocumentRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	DocumentRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<LayerPage>. Await the Task.

Property	Type
Document	DocumentInfo
Layers	LayerInfo[]
LinePatterns	string[]
LineWidths	string[]

Request record

```
public sealed record DocumentRequest(string DocumentId, long ExpectedRevision);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadAutomationApi.ExecuteOperationAsync

Execute an immediate operation — Executes an immediate operation as one Undo step. Send worker operations to StartOperationAsync. Unknown parameter fields are rejected.

Declaration

```
Task<EditResult> ExecuteOperationAsync(

    OperationRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	OperationRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record OperationRequest(string DocumentId, long ExpectedRevision, string OperationId,

    string Operation, JsonElement Parameters, bool SelectResult = true);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

Drawing: lines, circles and closed profiles

DrawingAsync creates a 60 mm line and radius-10 circle in one Undo step. RectangleAsync creates a closed 80 × 50 polyline; do not repeat the first vertex when Closed=true. Pass returned entity IDs to later edits.

Working example: drawing

```
public static async Task<EditResult> DrawingAsync(
     ICadApi cad, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     var request = new CreateGeometryRequest(doc.Id, doc.Revision,
         Guid.NewGuid().ToString("D"),
         new[] { new LineDefinition(new(0, 0, 0), new(60, 0, 0)) },
         new[] { new CircleDefinition(new(30, 20, 0), 10, new(0, 0, 1)) });
     EditResult result = await cad.CreateGeometryAsync(request, ct);
     Console.WriteLine($"Created: {result.Entities.Length}; revision: {result.Document.Revision}");
     return result;
 }
```

Working example: curves

```
public static async Task<EditResult> RectangleAsync(
     ICadModelApi cad, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     var rectangle = new PolylineDefinition(new Point3[] {
         new(0, 0, 0), new(80, 0, 0), new(80, 50, 0), new(0, 50, 0)
     }, Closed: true);
     return await cad.CreateCurvesAsync(new CreateCurvesRequest(
         doc.Id, doc.Revision, Guid.NewGuid().ToString("D"),
         new[] { rectangle }, Array.Empty<ArcDefinition>()), ct);
 }
```

ICadApi.CreateGeometryAsync

Create lines and circles — Creates 1–1000 lines/circles in one atomic Undo batch. Pass empty arrays for unused kinds and a nonzero circle normal.

Declaration

```
Task<EditResult> CreateGeometryAsync(

    CreateGeometryRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	CreateGeometryRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]

Property	Type
Changed	bool

Request record

```
public sealed record CreateGeometryRequest(string DocumentId, long ExpectedRevision, string OperationId,

    LineDefinition[] Lines, CircleDefinition[] Circles, bool SelectCreated = true);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadModelApi.CreateCurvesAsync

Create polylines and arcs — Creates up to 1000 curves and 4096 polyline vertices. Do not repeat the first point to close a polyline. Arc sweeps are signed right-hand angles with 0 < absolute sweep < 360 degrees.

Declaration

```
Task<EditResult> CreateCurvesAsync(

    CreateCurvesRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	CreateCurvesRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record CreateCurvesRequest(string DocumentId, long ExpectedRevision, string OperationId,

    PolylineDefinition[] Polylines, ArcDefinition[] Arcs, bool SelectCreated = true);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does

not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

Selection and geometry inspection

Selection results may be paged; check Total and Offset. Inspection paging must use the same revision. Refresh face and edge references after edits; do not reuse a hard-coded subobject index on another shape.

Working example: selection

```
public static async Task<SelectionInfo> SelectCreatedAsync(
     ICadModelApi cad, CancellationToken ct = default)
 {
     EditResult created = await DrawingAsync(cad, ct);
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     await cad.SetSelectionAsync(new EntityEditRequest(doc.Id, doc.Revision,
         Guid.NewGuid().ToString("D"), created.Entities.Select(e => e.Id).ToArray()), ct);
     SelectionInfo selection = await cad.GetSelectionAsync(0, 256, ct);
     Console.WriteLine($"Selected {selection.Entities.Length} of {selection.Total}");
     return selection;
 }
```

Working example: inspect

```
public static async Task InspectAsync(
     ICadAutomationApi cad, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     int offset = 0;
     while (true)
     {
         EntityPage page = await cad.GetEntitiesAsync(
             new EntityQuery(doc.Id, doc.Revision, offset, 100), ct);
         if (page.Entities.Length == 0) break;
         var request = new EntityDetailsRequest(doc.Id, doc.Revision,
             page.Entities.Select(e => e.Id).ToArray());
         EntityDetailsResult details = await cad.GetEntityDetailsAsync(request, ct);
         AttributePage attributes = await cad.GetAttributesAsync(request, ct);
         foreach (EntityDetail item in details.Entities)
             Console.WriteLine($"{item.Entity.Id}: {item.GeometryStatus}");
         foreach (AttributeInfo item in attributes.Items)
             Console.WriteLine($"Layer: {item.Entity.Layer}; locked: {item.LayerLocked}");
         offset += page.Entities.Length;
         if (offset >= page.Total) break;
     }
 }
```

Working example: topology

```
public static async Task<TopologyItem[]> EdgesAsync(
     ICadAutomationApi cad, string entityId, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     var edges = new List<TopologyItem>();
     do
     {
         TopologyPage page = await cad.GetTopologyAsync(new TopologyRequest(
             doc.Id, doc.Revision, entityId, "edge", edges.Count, 128), ct);
         edges.AddRange(page.Items);
         if (edges.Count >= page.Total || page.Items.Length == 0) break;
     } while (true);
     return edges.ToArray();
 }
```

ICadApi.GetSelectionAsync

Read the selection — Pages through selected entities with IDs, types, layers and WCS bounds. Offset is zero-based; limit is 1–1000.

Declaration

```
Task<SelectionInfo> GetSelectionAsync(

    int offset = 0,
```

```
    int limit = 256,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
offset	int	Number of entries to skip; zero-based.
limit	int	Maximum items per page within this method’s limit.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<SelectionInfo>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Total	int
Offset	int

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadModelApi.GetEntitiesAsync

List model entities — Lists top-level model entities in drawing order. Keep the same revision across pages; restart paging if the drawing changes.

Declaration

```
Task<EntityPage> GetEntitiesAsync(

    EntityQuery request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	EntityQuery	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EntityPage>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Total	int
Offset	int

Request record

```
public sealed record EntityQuery(string DocumentId, long ExpectedRevision, int Offset = 0, int Limit = 256);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadModelApi.GetEntityDetailsAsync

Read basic entity geometry — Reads exact line, circle, arc and linear polyline definitions for 1–100 IDs. Inspect GeometryStatus: unsupported/too_large does not mean geometry was returned.

Declaration

```
Task<EntityDetailsResult> GetEntityDetailsAsync(

    EntityDetailsRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	EntityDetailsRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EntityDetailsResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityDetail[]

Request record

```
public sealed record EntityDetailsRequest(string DocumentId, long ExpectedRevision, string[] EntityIds);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadModelApi.SetSelectionAsync

Replace the selection — Selects explicit top-level IDs; an empty array clears selection. Changes highlighting without adding an Undo step.

Declaration

```
Task<EditResult> SetSelectionAsync(

    EntityEditRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	EntityEditRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record EntityEditRequest(string DocumentId, long ExpectedRevision, string OperationId, string[]
EntityIds);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadAutomationApi.GetAttributesAsync

Read entity attributes — Reads color, line pattern, line width and layer lock for 1–100 IDs. Distinguishes byLayer/byBlock/foreground/rgb colors.

Declaration

```
Task<AttributePage> GetAttributesAsync(

    EntityDetailsRequest request,
```

```
    CancellationToken cancellationToken = default
);
```

Parameter	Type	Meaning
request	EntityDetailsRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<AttributePage>. Await the Task.

Property	Type
Document	DocumentInfo
Items	AttributeInfo[]

Request record

```
public sealed record EntityDetailsRequest(string DocumentId, long ExpectedRevision, string[] EntityIds);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadAutomationApi.GetTopologyAsync

Read face edge and vertex references — Pages through faces, edges or vertices of a solid, shell or face. Kind is face/edge/vertex; limit is 1–256. SubobjectRef expires on revision changes.

Declaration

```
Task<TopologyPage> GetTopologyAsync(

    TopologyRequest request,

    CancellationToken cancellationToken = default
);
```

Parameter	Type	Meaning
request	TopologyRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<TopologyPage>. Await the Task.

Property	Type
Document	DocumentInfo
Items	TopologyItem[]
Total	int
Offset	int

Request record

```
public sealed record TopologyRequest(string DocumentId, long ExpectedRevision, string EntityId,

    string Kind = "face", int Offset = 0, int Limit = 128);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadWorkspaceApi.QueryAsync

Query extended drawing data — Kind geometry reads extended geometry; sheet reads sheet-metal data; blocks lists definitions/direct model insertions; styles lists style names. Use 1–100 IDs for geometry/sheet and an empty array otherwise.

Declaration

```
Task<QueryResult> QueryAsync(

    QueryRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	QueryRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<QueryResult>. Await the Task.

Property	Type
Document	DocumentInfo
Data	Text.Json.JsonElement

Request record

```
public sealed record QueryRequest(string DocumentId,long ExpectedRevision,string Kind,string[] EntityIds);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

Input and interactive commands

Receive input in a custom command

ICadInputApi returns user input without launching a standard drawing command. Await GetPointAsync or another input method, check Accepted, then use Point or Entities to create/edit geometry. Never use .Wait()/.Result on the UI thread. Carry the returned Document into the next request.

Esc returns cancelled; explicitly permitted empty input returns none. CancellationToken cancellation throws OperationCanceledException. CursorPoint is provisional. Changing the document, view or geometry cancels input. Transient previews are not saved and create no Undo entry.

Coordinates are WCS millimetres and angles are degrees. Angles and rectangles use the current drawing plane. Drag left-to-right for window selection and right-to-left for crossing selection. Type/layer lists use OR within each list and AND across lists. Refresh face/edge references after geometry changes.

Complete projects are in ApiExamples/InputCS and InputVB, and Customization/SDK/Samples/Customization.InputCS and Customization.InputVB. They provide two-point lines, circle previews and selection-copy commands. Loading custom DLLs requires STD/Pro.

Method	Use and result
GetPointAsync	Pick a point; passing a base point enables rubber-band feedback. Uses current snapping.
GetCornerAsync	Preview a rectangle in the current drawing plane and return its opposite corner.
GetDistanceAsync / GetAngleAsync	Use a base point and cursor, or numeric entry. Number is in mm/degrees.
GetIntegerAsync / GetDoubleAsync	Enter integer/real values with defaults and optional bounds.
GetStringAsync / GetKeywordsAsync	Enter text/a keyword. Text contains the result; keywords allow unique prefixes.
GetEntityAsync / GetSelectionAsync	Pick one/multiple entities. Enter accepts; Kinds/Layers filter results.
GetSubentityAsync	Pick face/edge; Subobjects contains topology references.
SelectRegionAsync	all/window/crossing selection; observe visibility and locking options.
DragAsync / UpdateInputPreviewAsync	Preview up to 256 lines/circles from the provisional cursor position.
StartInputAsync / GetInputAsync / CancelInputAsync	Separate start, poll and cancel calls for MCP and other clients.

Create a line from two points

The following code implements IAddinCommand. In C#, import System, System.Threading, System.Threading.Tasks and JDesignCAD.Api with using; in VB.NET, use Imports for the same namespaces. The supplied projects contain the complete namespace, registration code and addin.json.

C#

```
public sealed class TwoPointLineCommand : IAddinCommand
{
    public async Task<CommandResult> ExecuteAsync(ICadApi api,CancellationToken cancellationToken)
    {
        var input=(ICadInputApi)api;
        var doc=await api.GetDocumentAsync(cancellationToken);

        // 1 点目をユーザーが指示するまで待つ。
        var first=await input.GetPointAsync(doc,"1 点目を指示してください",cancellationToken:cancellationToken);
        if(!first.Accepted)return new CommandResult("");

        // 1 点目からマウス位置へラバーバンドを表示し、2 点目を待つ。
```

```
        var second=await input.GetPointAsync(first.Document,"2 点目を指示してください",
            basePoint:first.Point,cancellationToken:cancellationToken);
        if(!second.Accepted)return new CommandResult("");

        var a=first.Point!;var b=second.Point!;
        if(Math.Sqrt(Math.Pow(b.X-a.X,2)+Math.Pow(b.Y-a.Y,2)+Math.Pow(b.Z-a.Z,2))<=1e-6)
            return new CommandResult("異なる 2 点を指示してください。");

        // ここからが自作の作図処理。標準の線分コマンドは呼ばない。
        await api.CreateGeometryAsync(new CreateGeometryRequest(
            second.Document.Id,second.Document.Revision,Guid.NewGuid().ToString("D"),
            new[]{new LineDefinition(a,b)},Array.Empty<CircleDefinition>()),cancellationToken);
        return new CommandResult("");
    }
}
```

VB.NET

```
Public NotInheritable Class TwoPointLineCommand
        Implements IAddinCommand
        Public Async Function ExecuteAsync(api As ICadApi, cancellationToken As CancellationToken) As Task(Of
CommandResult) Implements IAddinCommand.ExecuteAsync
            Dim input = DirectCast(api, ICadInputApi)
            Dim doc = Await api.GetDocumentAsync(cancellationToken)

            ' 1 点目をユーザーが指示するまで待つ。
            Dim first = Await input.GetPointAsync(doc, "1 点目を指示してください", cancellationToken:=cancellationToken)
            If Not first.Accepted Then Return New CommandResult("")

            ' 1 点目からマウス位置へラバーバンドを表示し、2 点目を待つ。
            Dim second = Await input.GetPointAsync(first.Document, "2 点目を指示してください", basePoint:=first.Point,
cancellationToken:=cancellationToken)
            If Not second.Accepted Then Return New CommandResult("")
            Dim a = first.Point
            Dim b = second.Point
            If Math.Sqrt((b.X - a.X) ^ 2 + (b.Y - a.Y) ^ 2 + (b.Z - a.Z) ^ 2) <= 0.000001 Then Return New
CommandResult("異なる 2 点を指示してください。")

            ' ここからが自作の作図処理。標準の線分コマンドは呼ばない。
            Await api.CreateGeometryAsync(New CreateGeometryRequest(
                second.Document.Id, second.Document.Revision, Guid.NewGuid().ToString("D"),
                New LineDefinition() {New LineDefinition(a, b)}, Array.Empty(Of CircleDefinition)()),
cancellationToken)
            Return New CommandResult("")
        End Function
    End Class
```

Nested block picks, fence/polygon selection, keyword branching during point input and arbitrary 3D-geometry Jigs are not exposed. This is not a binary-compatible AutoCAD API.

In MCP, use cad_start_input to start and cad_get_input to poll. cad_cancel_input cancels, cad_update_input_preview updates feedback, and cad_select_region selects a region. waiting is not completion: only accepted contains confirmed input.

ICadInputApi returns user input without launching a standard drawing command. Await GetPointAsync or another input method, check Accepted, then use Point or Entities to create/edit geometry. Never use .Wait()/.Result on the UI thread. Carry the returned Document into the next request.

Working example: input

```
public static async Task<InteractiveCommandResult> LaunchInputAsync(
     ICadInteractionApi cad, string commandId,
     CancellationToken ct = default)
 {
     InteractiveCommandInfo[] commands = await cad.GetInteractiveCommandsAsync(ct);
     InteractiveCommandInfo command = commands.Single(c => c.Id == commandId);
     if (!command.Available)
```

```
        throw new InvalidOperationException(command.UnavailableReason);
    DocumentInfo doc = await cad.GetDocumentAsync(ct);
    return await cad.LaunchInteractiveCommandAsync(new InteractiveCommandRequest(
        doc.Id, doc.Revision, Guid.NewGuid().ToString("D"), command.Id), ct);
}
```

ICadInteractionApi.GetInteractiveCommandsAsync

List interactive GUI commands — Lists 266 registered GUI command IDs and their availability. These launch user interactions; they are not parameterized geometry APIs.

Declaration

```
Task<InteractiveCommandInfo[]> GetInteractiveCommandsAsync(

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<InteractiveCommandInfo[]>. Await the Task.

Property	Type
Id	string
Group	string
Available	bool
UnavailableReason	string
UnavailableCode	string
LicenseRestriction	LicenseRestriction

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadInteractionApi.LaunchInteractiveCommandAsync

Queue an interactive GUI command — Queues a GUI command for the user to complete. queued acknowledges dispatch; it does not confirm geometry creation, saving or analysis completion.

Declaration

```
Task<InteractiveCommandResult> LaunchInteractiveCommandAsync(

    InteractiveCommandRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	InteractiveCommandRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<InteractiveCommandResult>. Await the Task.

Property	Type
CommandId	string
State	string
Message	string

Request record

```
public sealed record InteractiveCommandRequest(string DocumentId,long ExpectedRevision,string OperationId,string CommandId);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

Editing, copying and history

CopySelectionAsync makes a copy 100 mm along X and preserves the source. The delete/history example targets only its own newly created objects. Do not interleave unrelated user edits with Undo/Redo; carry the returned document revision forward.

Working example: edit

```
public static async Task<EditResult> CopySelectionAsync(
     ICadModelApi cad, CancellationToken ct = default)
 {
     SelectionInfo selected = await cad.GetSelectionAsync(0, 256, ct);
     if (selected.Total == 0 || selected.Total > selected.Entities.Length)
         throw new InvalidOperationException("Select 1 to 256 complete model objects.");
     DocumentInfo doc = selected.Document;
     return await cad.TranslateEntitiesAsync(new TranslateEntitiesRequest(
         doc.Id, doc.Revision, Guid.NewGuid().ToString("D"),
         selected.Entities.Select(e => e.Id).ToArray(),
         new Point3(100, 0, 0), Copy: true), ct);
 }
```

Working example: history

```
public static async Task DeleteOwnExampleAndUndoAsync(
     ICadModelApi cad, CancellationToken ct = default)
 {
     EditResult made = await DrawingAsync(cad, ct);
     DocumentInfo doc = made.Document;
     EditResult deleted = await cad.DeleteEntitiesAsync(new EntityEditRequest(
         doc.Id, doc.Revision, Guid.NewGuid().ToString("D"),
         made.Entities.Select(e => e.Id).ToArray()), ct);
     doc = deleted.Document;
     EditResult undone = await cad.UndoAsync(new HistoryRequest(
         doc.Id, doc.Revision, Guid.NewGuid().ToString("D")), ct);
     doc = undone.Document;
     if (doc.CanRedo)
         await cad.RedoAsync(new HistoryRequest(
             doc.Id, doc.Revision, Guid.NewGuid().ToString("D")), ct);
 }
```

ICadModelApi.TranslateEntitiesAsync

Move or copy entities — Translates 1–1000 top-level entities by WCS displacement. Copy=true preserves originals and their attributes; the entire batch is one Undo step.

Declaration

```
Task<EditResult> TranslateEntitiesAsync(

    TranslateEntitiesRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	TranslateEntitiesRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record TranslateEntitiesRequest(string DocumentId, long ExpectedRevision, string OperationId,

    string[] EntityIds, Point3 Displacement, bool Copy = false, bool SelectResult = true);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadModelApi.DeleteEntitiesAsync

Delete entities — Deletes 1–1000 explicit IDs after inspection. Locked layers and stale/removed IDs are rejected.

Declaration

```
Task<EditResult> DeleteEntitiesAsync(

    EntityEditRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	EntityEditRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record EntityEditRequest(string DocumentId, long ExpectedRevision, string OperationId, string[]
EntityIds);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadApi.UndoAsync

Undo the latest operation — Undoes the latest drawing operation, including manual edits. This is not restricted to edits made by the API.

Declaration

```
Task<EditResult> UndoAsync(

    HistoryRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	HistoryRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record HistoryRequest(string DocumentId, long ExpectedRevision, string OperationId);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadApi.RedoAsync

Redo the latest undone operation — Check CanRedo first. The returned Document describes the resulting drawing state.

Declaration

```
Task<EditResult> RedoAsync(

    HistoryRequest request,

    CancellationToken cancellationToken = default
```

);

Parameter	Type	Meaning
request	HistoryRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record HistoryRequest(string DocumentId, long ExpectedRevision, string OperationId);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

3D and background operations

RunEditAsync implements start, polling, ready and commit. A drawing change during calculation invalidates the result; refresh selections and references before restarting. Cancellation also calls CancelJobAsync. BoxAsync creates an 80 × 50 × 20 mm solid.

Working example: worker

```
public static async Task<EditResult> RunEditAsync<T>(
     ICadJobApi cad, string operation, T parameters,
     CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     JobInfo job = await cad.StartOperationAsync(new StartOperationRequest(
         doc.Operation(operation, parameters), TimeoutSeconds: 120), ct);
     try
     {
         while (job.State == "running")
         {
             await Task.Delay(200, ct);
             job = await cad.GetJobAsync(job.Id, ct);
         }
         if (job.State != "ready")
             throw new InvalidOperationException($"{job.State}: {job.ErrorCode}: {job.Message}");
         return await cad.CommitJobAsync(new CommitJobRequest(
             doc.Id, doc.Revision, Guid.NewGuid().ToString("D"), job.Id), ct);
     }
     catch (OperationCanceledException)
     {
         await cad.CancelJobAsync(job.Id, CancellationToken.None);
         throw;
     }
 }
```

Working example: solid

```
public static Task<EditResult> BoxAsync(
     ICadJobApi cad, CancellationToken ct = default) =>
     RunEditAsync(cad, "solid.primitive",
         new PrimitiveParameters("box", new(0, 0, 0),
             Size: new Point3(80, 50, 20)), ct);
```

ICadJobApi.StartOperationAsync

Start an isolated calculation — Calculates detached geometry in a separate worker process without editing the drawing. At most four pending running/ready jobs and one active worker; timeout is 1–1800 seconds.

Declaration

```
Task<JobInfo> StartOperationAsync(

    StartOperationRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	StartOperationRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<JobInfo>. Await the Task.

Property	Type
Id	string
DocumentId	string
SourceRevision	long
Operation	string
State	string
Stage	string
ResultCount	int
ErrorCode	string
Message	string

Request record

```
public sealed record StartOperationRequest(OperationRequest Request, int TimeoutSeconds = 120);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadJobApi.GetJobAsync

Read calculation state — Distinguishes running, ready, completed, failed, cancelled and committed. A ready result has not yet been applied to the drawing.

Declaration

```
Task<JobInfo> GetJobAsync(

    string jobId,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
jobId	string	JobInfo.Id returned by StartOperationAsync.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<JobInfo>. Await the Task.

Property	Type
Id	string

Property	Type
DocumentId	string
SourceRevision	long
Operation	string
State	string
Stage	string
ResultCount	int
ErrorCode	string
Message	string

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadJobApi.CancelJobAsync

Cancel a job or discard a ready result — Cancels running work or discards ready geometry. Use UndoAsync to undo a committed edit.

Declaration

```
Task<JobInfo> CancelJobAsync(

    string jobId,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
jobId	string	JobInfo.Id returned by StartOperationAsync.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<JobInfo>. Await the Task.

Property	Type
Id	string
DocumentId	string
SourceRevision	long
Operation	string
State	string

Property	Type
Stage	string
ResultCount	int
ErrorCode	string
Message	string

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

ICadJobApi.CommitJobAsync

Commit a ready result — Commits a ready result in one Undo step. Rejects drawing changes since calculation started and rechecks permissions. Use a new commit UUID, separate from the start UUID.

Declaration

```
Task<EditResult> CommitJobAsync(

    CommitJobRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	CommitJobRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<EditResult>. Await the Task.

Property	Type
Document	DocumentInfo
Entities	EntityInfo[]
Changed	bool

Request record

```
public sealed record CommitJobRequest(string DocumentId, long ExpectedRevision, string OperationId, string JobId);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does

not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

Sheet metal

SheetAsync first displays the project defaults, then creates an example plate with thickness 1, inner radius 2 and K factor 0.4. These example values are not universal manufacturing defaults. Obtain fresh edge references before adding flanges.

Working example: sheet

```
public static async Task<EditResult> SheetAsync(
     ICadWorkspaceApi cad, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     QueryResult defaults = await cad.QueryAsync(new QueryRequest(
         doc.Id, doc.Revision, "sheetDefaults", Array.Empty<string>()), ct);
     Console.WriteLine(defaults.Data.GetRawText());
     EditResult profile = await RectangleAsync(cad, ct);
     // This example intentionally specifies t=1, R=2, K=0.4.
     // Use the drawing or the inspected project defaults for production parts.
     return await RunEditAsync(cad, "sheet.plate", new SheetPlateParameters(
         profile.Entities.Single().Id, new SheetRules(1, 2, .4)), ct);
 }
```

Measurement and analysis

Measurement/analysis operations return worker-data, retrieved with GetJobDataAsync after completed. AnalysisAsync is a single-material example. Assemblies use analysis.assembly.contacts/mesh/solve with per-body materials. Retrieve all pages of points, elements and boundaries before solving.

Working example: data

```
public static async Task<JobDataPage> MeasureAsync(
     ICadWorkspaceApi cad, string[] entityIds, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     JobInfo job = await cad.StartOperationAsync(new StartOperationRequest(
         doc.Operation("measure.geometry", new MeasureParameters(entityIds))), ct);
     while (job.State == "running")
     {
         await Task.Delay(200, ct);
         job = await cad.GetJobAsync(job.Id, ct);
     }
     if (job.State != "completed")
         throw new InvalidOperationException($"{job.State}: {job.Message}");
     return await cad.GetJobDataAsync(new JobDataRequest(job.Id, "summary"), ct);
 }
```

Working example: analysis

```
public static async Task<JobInfo> RunDataAsync<T>(
     ICadWorkspaceApi cad, string operation, T parameters,
     CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     JobInfo job = await cad.StartOperationAsync(new StartOperationRequest(
         doc.Operation(operation, parameters), TimeoutSeconds: 120), ct);
     while (job.State == "running")
     {
         await Task.Delay(200, ct);
         job = await cad.GetJobAsync(job.Id, ct);
     }
     if (job.State != "completed")
         throw new InvalidOperationException($"{job.State}: {job.Message}");
     return job;
 }
 public static async Task<T[]> ReadArrayAsync<T>(
     ICadWorkspaceApi cad, string jobId, string section,
     CancellationToken ct = default)
 {
     var items = new List<T>();
     do
     {
         JobDataPage page = await cad.GetJobDataAsync(
             new JobDataRequest(jobId, section, items.Count, 128), ct);
         T[] batch = page.Data.Deserialize<T[]>(LocalProtocol.Json)
             ?? throw new InvalidOperationException("Missing result data.");
         items.AddRange(batch);
         if (items.Count >= page.Total || batch.Length == 0) break;
     } while (true);
     return items.ToArray();
 }
 public static async Task<JobDataPage> AnalysisAsync(
     ICadWorkspaceApi cad, CancellationToken ct = default)
 {
     EditResult box = await BoxAsync(cad, ct);
     JobInfo job = await RunDataAsync(cad, "analysis.mesh",
         new AnalysisMeshParameters(new[] { box.Entities.Single().Id }, 3, 10), ct);
     double[][] points = await ReadArrayAsync<double[]>(cad, job.Id, "points", ct);
     int[][] elements = await ReadArrayAsync<int[]>(cad, job.Id, "elements", ct);
     AnalysisBoundaryData[] boundaries = await ReadArrayAsync<AnalysisBoundaryData>(
         cad, job.Id, "boundaries", ct);
     int[] GroupsAt(double x) => boundaries.Where(b =>
         b.Nodes.All(n => Math.Abs(points[n][0] - x) < 1e-6))
```

```
        .Select(b => b.Group).Distinct().ToArray();
    int[] fixedGroups = GroupsAt(0), loadedGroups = GroupsAt(80);
    if (fixedGroups.Length == 0 || loadedGroups.Length == 0)
        throw new InvalidOperationException("Expected end faces were not found.");
    var mesh = new AnalysisMeshData(3, points, elements, boundaries);
    var conditions = new[] {
        new AnalysisConditionData("fixed", Groups: fixedGroups),
        new AnalysisConditionData("totalforce", Groups: loadedGroups, Z: -100)
    };
    JobInfo result = await RunDataAsync(cad, "analysis.solve",
        new AnalysisSolveParameters(mesh, new AnalysisMaterialData(), conditions), ct);
    return await cad.GetJobDataAsync(new JobDataRequest(result.Id, "summary"), ct);
}
```

ICadWorkspaceApi.GetJobDataAsync

Read completed measurement or analysis data — Reads summary or a named result array from a completed data job, 1–256 items per page. Data jobs require no commit.

Declaration

```
Task<JobDataPage> GetJobDataAsync(

    JobDataRequest request,

    CancellationToken cancellationToken = default

);
```

Parameter	Type	Meaning
request	JobDataRequest	Construct the request type below. Editing requires current document/revision and a new UUID.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<JobDataPage>. Await the Task.

Property	Type
Job	JobInfo
Section	string
Data	Text.Json.JsonElement
Total	int
Offset	int

Request record

```
public sealed record JobDataRequest(string JobId,string Section="summary",int Offset=0,int Limit=128);
```

Errors and conditions

Check no_document, invalid_argument and document_changed/revision_conflict/entity_not_found as applicable. Editing may return cad_busy/layer_locked/undo_disabled. Restricted operations return license_required; reading a schema does not authorize execution. Jobs may return job_not_found/job_not_ready; calculations may fail with geometry_failed/invalid_result. See the shared error guide.

Layouts, annotations and preview

LayoutAsync places an independent top view on landscape A4. Paper coordinates are sheet millimetres; ModelOrigin is WCS model millimetres. Match VisibleLayers to the geometry and use a unique layout name. Add native leaders with layout.annotations and verify rendering with view.capture. This is distinct from interactive VIEWBASE/VIEWPROJ projection relationships.

Working example: layout

```
public static async Task<EditResult> LayoutAsync(
     ICadAutomationApi cad, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     var view = new LayoutViewParameters(new(0, 0, -1), new(0, 1, 0),
         new(40, 25, 0), new(100, 110, 0), 1,
         new LayoutRectangle(20, 180, 50, 180),
         new[] { "Standard", "0", "Example" });
     return await cad.ExecuteOperationAsync(doc.Operation("layout.create",
         new LayoutCreateParameters("Example-A4", 297, 210, new[] { view },
             Array.Empty<GeometryDefinition>(), Array.Empty<LayoutDimensionParameters>())), ct);
 }
```

Open, save and export

Use an absolute path and wait for completed; verify Bytes and Sha256. Overwrite is false by default. Save changes the drawing filename; Export writes a copy. Inspect Warnings. document.open requires a drawing with no unsaved changes. Geometry Undo does not undo file output.

Working example: files

```
public static async Task<FileOperationInfo> SaveAsync(
     ICadFileApi cad, string absolutePath, CancellationToken ct = default)
 {
     FileFormatInfo[] formats = await cad.GetFileFormatsAsync(ct);
     if (!formats.Any(f => f.Format == "jdc" && f.CanSave && f.Available))
         throw new InvalidOperationException("JDC saving is unavailable.");
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     FileOperationInfo operation = await cad.SaveDocumentAsync(new SaveDocumentRequest(
         doc.Id, doc.Revision, Guid.NewGuid().ToString("D"), absolutePath, "jdc"), ct);
     return await WaitFileAsync(cad, operation, ct);
 }
 public static async Task<FileOperationInfo> ExportStepAsync(
     ICadFileApi cad, string absolutePath, CancellationToken ct = default)
 {
     DocumentInfo doc = await cad.GetDocumentAsync(ct);
     FileOperationInfo operation = await cad.ExportDocumentAsync(new ExportDocumentRequest(
         doc.Id, doc.Revision, Guid.NewGuid().ToString("D"), absolutePath, "step"), ct);
     return await WaitFileAsync(cad, operation, ct);
 }
 public static async Task<FileOperationInfo> WaitFileAsync(
     ICadFileApi cad, FileOperationInfo operation, CancellationToken ct = default)
 {
     try
     {
         while (operation.State == "running")
         {
             await Task.Delay(200, ct);
             operation = await cad.GetFileOperationAsync(operation.OperationId, ct);
         }
     }
     catch (OperationCanceledException)
     {
         await cad.CancelFileOperationAsync(operation.OperationId, CancellationToken.None);
         throw;
     }
     if (operation.State != "completed")
         throw new InvalidOperationException($"{operation.State}: {operation.Error?.Message}");
     foreach (string warning in operation.Warnings ?? Array.Empty<string>())
         Console.WriteLine(warning);
     Console.WriteLine($"{operation.FilePath}: {operation.Bytes} bytes; {operation.Sha256}");
     return operation;
 }
```

ICadFileApi.GetFileFormatsAsync

```
Task<FileFormatInfo[]> GetFileFormatsAsync(CancellationToken cancellationToken = default);
```

The files example in this category shows complete request construction, status polling and result handling.

ICadFileApi.SaveDocumentAsync

```
Task<FileOperationInfo> SaveDocumentAsync(SaveDocumentRequest request, CancellationToken cancellationToken = default);
```

The files example in this category shows complete request construction, status polling and result handling.

ICadFileApi.ExportDocumentAsync

```
Task<FileOperationInfo> ExportDocumentAsync(ExportDocumentRequest request, CancellationToken cancellationToken =
default);
```

The files example in this category shows complete request construction, status polling and result handling.

ICadFileApi.GetFileOperationAsync

```
Task<FileOperationInfo> GetFileOperationAsync(string operationId, CancellationToken cancellationToken = default);
```

The files example in this category shows complete request construction, status polling and result handling.

ICadFileApi.CancelFileOperationAsync

```
Task<FileOperationInfo> CancelFileOperationAsync(string operationId, CancellationToken cancellationToken = default);
```

The files example in this category shows complete request construction, status polling and result handling.

Command registration and add-ins

Build ExampleAddin and Samples in the same project. OnStartup registers the ribbon; button ExecuteAsync performs drawing. OnShutdown should release your event subscriptions and timers; this sample has none. Custom add-ins require a supporting license.

Working example: addin

```
public sealed class ExampleAddin : ICadAddin
{
 public void OnStartup(IAddinContext context)
 {
     context.RegisterTab(new RibbonTab("example", "Example", new[] {
         new RibbonPanel("draw", "Draw", new[] {
             new RibbonButton("line-circle", "Line and circle",
                 "Create example geometry", new DrawCommand())
         })
     }));
 }
 public void OnShutdown() { }
 private sealed class DrawCommand : IAddinCommand
 {
     public async Task<CommandResult> ExecuteAsync(ICadApi api, CancellationToken ct)
     {
         EditResult result = await Samples.DrawingAsync(api, ct);
         return new CommandResult($"Created {result.Entities.Length} objects.");
     }
 }
}
```

IAddinCommand.ExecuteAsync

Implement an add-in button — Called by the ribbon button. Access the drawing through api and forward the cancellation token. The host displays a nonempty returned Message.

Declaration

```
Task<CommandResult> ExecuteAsync(

    ICadApi api,

    CancellationToken cancellationToken

);
```

Parameter	Type	Meaning
api	ICadApi	Host API; test and cast to the required extended interface.
cancellationToken	CancellationToken	Cancellation token; cancels queued work before it begins. A short atomic edit completes once started.

Return value

Task<CommandResult>. Await the Task.

Property	Type
Message	string

Tabs are installed only after successful startup registration. Free executes the built-in sample only; loading custom or modified DLLs requires STD/Pro.

IAddinContext.RegisterTab

Register a ribbon tab — Call only during OnStartup. IDs must be unique. Limits: 10 tabs/add-in, 1–20 panels/tab and 1–50 buttons/panel.

Declaration

```
void RegisterTab(

    RibbonTab tab

);
```

Parameter	Type	Meaning
tab	RibbonTab	RibbonTab containing panel and button definitions.

Return value

void

Tabs are installed only after successful startup registration. Free executes the built-in sample only; loading custom or modified DLLs requires STD/Pro.

ICadAddin.OnStartup

Register an add-in at startup — Called by the host after loading the add-in. Register tabs/panels/buttons through context.RegisterTab; defer calculations and drawing edits to commands.

Declaration

```
void OnStartup(

    IAddinContext context

);
```

Parameter	Type	Meaning
context	IAddinContext	Host registration context exposing Api and AddinDirectory.

Return value

void

Tabs are installed only after successful startup registration. Free executes the built-in sample only; loading custom or modified DLLs requires STD/Pro.

ICadAddin.OnShutdown

Release add-in resources — Called when the host disposes the add-in. Release owned resources and subscriptions; do not hide drawing-save operations here.

Declaration

```
void OnShutdown();
```

No parameters.

Return value

void

Tabs are installed only after successful startup registration. Free executes the built-in sample only; loading custom or modified DLLs requires STD/Pro.

Method index

CancelFileOperationAsync 506
CancelJobAsync 498
CommitJobAsync 499
CreateCurvesAsync 479
CreateGeometryAsync 478
DeleteEntitiesAsync 493
ExecuteAsync 507
ExecuteOperationAsync 476
ExportDocumentAsync 505
GetAttributesAsync 484
GetCapabilitiesAsync 474
GetDocumentAsync 473
GetEntitiesAsync 482
GetEntityDetailsAsync 483
GetFileFormatsAsync 505
GetFileOperationAsync 505
GetInteractiveCommandsAsync 490
GetJobAsync 497
GetJobDataAsync 503
GetLayersAsync 475
GetOperationSchemaAsync 474
GetSelectionAsync 481
GetTopologyAsync 485
LaunchInteractiveCommandAsync 490
OnShutdown 508
OnStartup 508
QueryAsync 486
RedoAsync 494
RegisterTab 508
SaveDocumentAsync 505
SetSelectionAsync 484
StartOperationAsync 496
TranslateEntitiesAsync 492
UndoAsync 494

Parameter reference for 93 operations

Set OperationRequest.Operation and Parameters. immediate uses ExecuteOperationAsync; worker uses StartOperationAsync then CommitJobAsync; worker-data uses StartOperationAsync then GetJobDataAsync. Required fields refer to JSON input. Consult record definitions below for .NET defaults and nested types.

Settings and attributes 511

Drawing and annotation 513

Editing, copying and blocks 515

3D and face editing 520

Sheet metal 529

Measurement and analysis 536

Layouts and display 540

Files 545

Settings and attributes — operations and parameters

styles.lineWidth Create, update or delete a line width

Parameter record: LineWidthParameters / immediate

JSON parameter	Required	Type
name	yes	string
width	yes	number
makeCurrent	optional	boolean
scale	optional	string

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("styles.lineWidth",
    new LineWidthParameters("0.5mm", 0.5, MakeCurrent: true), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

styles.linePattern Create, update or delete a line pattern

Parameter record: LinePatternParameters / immediate

JSON parameter	Required	Type
name	yes	string
pattern	yes	array of number
makeCurrent	optional	boolean
scale	optional	string

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("styles.linePattern",
var result = await cad.ExecuteOperationAsync(request, ct);
```

attributes.set Set entity attributes

Parameter record: SetAttributesParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
layer	optional	string / null
color	optional	object / null
linePattern	optional	string / null
lineWidth	optional	string / null

```
public sealed record SetAttributesParameters(string[] EntityIds, string? Layer = null, CadColor? Color = null,
    string? LinePattern = null, string? LineWidth = null);
```

layers.delete Delete an unused layer

Parameter record: DeleteLayerParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
name	yes	string

```
public sealed record DeleteLayerParameters(string Name);
```

layers.set Create or edit a layer

Parameter record: SetLayerParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
name	yes	string
color	optional	object / null
locked	optional	boolean / null
transparency	optional	integer / null
makeCurrent	optional	boolean
linePattern	optional	string / null
lineWidth	optional	string / null

```
public sealed record SetLayerParameters(string Name, CadColor? Color = null, bool? Locked = null,
    int? Transparency = null, bool MakeCurrent = false, string? LinePattern = null, string? LineWidth = null);
```

Drawing and annotation — operations and parameters

annotation.dimension Create a dimension

Parameter record: DimensionParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
kind	yes	string
points	yes	array of object
textPosition	yes	object
normal	yes	object
direction	yes	object
radius	optional	number
style	optional	string / null

```
public sealed record DimensionParameters(string Kind,Point3[] Points,Point3 TextPosition,Point3 Normal,Point3
Direction,double Radius=0,string? Style=null);
```

annotation.hatch Create a hatch

Parameter record: HatchParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
profileId	yes	string
style	optional	string / null

```
public sealed record HatchParameters(string ProfileId,string? Style=null);
```

fill.create Create solid or gradient fills

Parameter record: CreateFillsParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
fills	yes	array of object

```
public sealed record CreateFillsParameters(FillDefinition[] Fills);
```

geometry.create Create geometry

Parameter record: CreateObjectsParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
objects	yes	array of object

```
public sealed record CreateObjectsParameters(GeometryDefinition[] Objects);
```

image.import Import a raster image

Parameter record: ImportImageParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
filePath	yes	string
location	yes	object
widthAxis	yes	object
heightAxis	yes	object

```
public sealed record ImportImageParameters(string FilePath, Point3 Location, Point3 WidthAxis, Point3 HeightAxis);
```

Editing, copying and blocks — operations and parameters

block.attributes Set block attribute values

Parameter record: BlockAttributesParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityId	yes	string
values	yes	array of object

```
public sealed record BlockAttributesParameters(string EntityId,BlockValue[] Values);
```

block.create Create a block definition and reference

Parameter record: BlockCreateParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
name	yes	string
basePoint	yes	object

```
public sealed record BlockCreateParameters(string[] EntityIds,string Name,Point3 BasePoint);
```

block.dynamic Change an existing dynamic value

Parameter record: BlockDynamicParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityId	yes	string
key	yes	string
value	yes	string

```
public sealed record BlockDynamicParameters(string EntityId,string Key,string Value);
```

block.insert Insert a block

Parameter record: BlockInsertParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
name	yes	string
position	yes	object
scale	optional	object / null
rotationDegrees	optional	number
normal	optional	object / null

```
public sealed record BlockInsertParameters(string Name,Point3 Position,Point3? Scale=null,double
RotationDegrees=0,Point3? Normal=null);
```

curve.join Join curves into one path

Parameter record: CurveJoinParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
tolerance	optional	number

```
public sealed record CurveJoinParameters(string[] EntityIds,double Tolerance=0.000001);
```

curve.offset Offset curves

Parameter record: CurveOffsetParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityId	yes	string
planeNormal	yes	object
distance	yes	number
tolerance	optional	number

```
public sealed record CurveOffsetParameters(string EntityId,Point3 PlaneNormal,double Distance,double Tolerance=0.001);
```

curve.split Split a curve

Parameter record: CurveSplitParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityId	yes	string
parameter	yes	number

```
public sealed record CurveSplitParameters(string EntityId,double Parameter);
```

curve.trim Trim a curve by parameters

Parameter record: TrimCurveParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityId	yes	string
startParameter	yes	number
endParameter	yes	number

```
public sealed record TrimCurveParameters(string EntityId, double StartParameter, double EndParameter);
```

entities.array.circular Circular array

Parameter record: CircularArrayParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
center	yes	object
axis	yes	object
stepDegrees	yes	number
count	yes	integer

```
public sealed record CircularArrayParameters(string[] EntityIds, Point3 Center, Point3 Axis, double StepDegrees, int Count);
```

entities.array.linear Linear array

Parameter record: LinearArrayParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
step	yes	object
count	yes	integer
includeOriginalCopy	optional	boolean

```
public sealed record LinearArrayParameters(string[] EntityIds, Point3 Step, int Count, bool IncludeOriginalCopy = false);
```

entities.explode Explode one level

Parameter record: ExplodeParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string

```
public sealed record ExplodeParameters(string[] EntityIds);
```

entities.mirror Mirror in a plane

Parameter record: MirrorParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
planeOrigin	yes	object
planeNormal	yes	object

JSON parameter	Required	Type
copy	optional	boolean

```
public sealed record MirrorParameters(string[] EntityIds, Point3 PlaneOrigin, Point3 PlaneNormal, bool Copy = false);
```

entities.rotate Rotate or rotate-copy

Parameter record: RotateParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
center	yes	object
axis	yes	object
angleDegrees	yes	number
copy	optional	boolean

```
public sealed record RotateParameters(string[] EntityIds, Point3 Center, Point3 Axis, double AngleDegrees, bool Copy =
false);
```

entities.scale Uniform scale

Parameter record: ScaleParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
center	yes	object
factor	yes	number
copy	optional	boolean

```
public sealed record ScaleParameters(string[] EntityIds, Point3 Center, double Factor, bool Copy = false);
```

fill.set Change fill appearance

Parameter record: SetFillParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
appearance	yes	object

```
public sealed record SetFillParameters(string[] EntityIds, FillAppearance Appearance);
```

geometry.update Update geometry

Parameter record: UpdateGeometryParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
updates	yes	array of object

```
public sealed record UpdateGeometryParameters(GeometryUpdate[] Updates);
```

text.update Update text

Parameter record: UpdateTextParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
updates	yes	array of object

```
public sealed record UpdateTextParameters(TextUpdate[] Updates);
```

3D and face editing — operations and parameters

face.label Add edit or remove a surface label

Parameter record: SurfaceLabelParameters / immediate / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
face	yes	object
center	yes	object
width	optional	number
height	optional	number
rotationDegrees	optional	number
text	optional	string / null
imagePath	optional	string / null
fontName	optional	string
color	optional	string
mode	optional	string
labelId	optional	string / null

```
public sealed record SurfaceLabelParameters(SubobjectRef Face, Point3 Center, double Width=30, double Height=10,
    double RotationDegrees=0, string? Text=null, string? ImagePath=null, string FontName="Yu Gothic",
    string Color="000000", string Mode="add", string? LabelId=null);
```

face.move Move selected faces

Parameter record: FaceMoveParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object
displacement	yes	object
moveConnected	optional	boolean

```
public sealed record FaceMoveParameters(string EntityId,SubobjectRef[] Faces,Point3 Displacement,bool
MoveConnected=false);
```

face.offset Offset selected faces

Parameter record: FaceOffsetParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object
distance	yes	number

```
public sealed record FaceOffsetParameters(string EntityId, SubobjectRef[] Faces, double Distance);
```

face.radius Change a round radius

Parameter record: FaceRadiusParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object
radius	yes	number

```
public sealed record FaceRadiusParameters(string EntityId,SubobjectRef[] Faces,double Radius);
```

face.remove Remove selected faces

Parameter record: RemoveFacesParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object

```
public sealed record RemoveFacesParameters(string EntityId, SubobjectRef[] Faces);
```

face.rotate Rotate selected faces

Parameter record: FaceRotateParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object
origin	yes	object
axis	yes	object
angleDegrees	yes	number
moveConnected	optional	boolean

```
public sealed record FaceRotateParameters(string EntityId,SubobjectRef[] Faces,Point3 Origin,Point3 Axis,double
AngleDegrees,bool MoveConnected=false);
```

geometry.extrude Extrude a profile

Parameter record: ExtrudeParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
direction	yes	object
solid	optional	boolean
deleteSources	optional	boolean

```
public sealed record ExtrudeParameters(string[] EntityIds, Point3 Direction, bool Solid = true, bool DeleteSources =
false);
```

geometry.clothoid Create a clothoid from curve parameters

Parameter record: ClothoidParameters / immediate

JSON parameter	Required	Type
first	yes	object
direction	yes	object
normal	yes	object
parameter	yes	number
startLength	yes	number
endLength	yes	number
clockwise	optional	boolean
chordDirection	optional	boolean
entityId	optional	string / null

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("geometry.clothoid",
    new ClothoidParameters(new(0, 0, 0), new(1, 0, 0), new(0, 0, 1),
        Parameter: 100, StartLength: 0, EndLength: 100), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

geometry.clothoid.basic Create a clothoid from basic design parameters

Parameter record: ClothoidBasicParameters / immediate

JSON parameter	Required	Type
entry	yes	object
intersection	yes	object
exit	yes	object
normal	yes	object
parameter	yes	number
radius	yes	number

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("geometry.clothoid.basic",
    new ClothoidBasicParameters(new(-100, 0, 0), new(0, 0, 0), new(0, 100, 0),
        new(0, 0, 1), Parameter: 20, Radius: 30), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

geometry.clothoid.stations Evaluate clothoid stations

Parameter record: ClothoidStationParameters / immediate

JSON parameter	Required	Type
entityId	yes	string
start	yes	number
left	yes	number
right	yes	number
normal	yes	object
interval	optional	number / null
tangent	optional	boolean

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
```

```
var selection = await cad.GetSelectionAsync(cancellationToken: ct);
string[] ids = selection.Entities.Select(e => e.Id).ToArray();
var request = doc.Operation("geometry.clothoid.stations",
    new ClothoidStationParameters(ids[0], Start: 0, Left: 3, Right: 3,
        Normal: new(0, 0, 1), Interval: 10), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

geometry.helix Create a helix

Parameter record: HelixParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
center	yes	object
axis	yes	object
baseRadius	yes	number
topRadius	yes	number
height	yes	number
turns	yes	number
segmentsPerTurn	optional	integer
clockwise	optional	boolean
startDirection	optional	object / null

```
public sealed record HelixParameters(Point3 Center, Point3 Axis, double BaseRadius, double TopRadius,
    double Height, double Turns, int SegmentsPerTurn = 48, bool Clockwise = false, Point3? StartDirection = null);
```

geometry.loft Loft through profiles

Parameter record: LoftParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
profileIds	yes	array of string
solid	optional	boolean
guideId	optional	string / null
deleteSources	optional	boolean

```
public sealed record LoftParameters(string[] ProfileIds, bool Solid = true, string? GuideId = null, bool DeleteSources
= false);
```

geometry.revolve Revolve a profile

Parameter record: RevolveParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityId	yes	string
origin	yes	object
axis	yes	object

JSON parameter	Required	Type
angleDegrees	optional	number
deleteSource	optional	boolean

```
public sealed record RevolveParameters(string EntityId, Point3 Origin, Point3 Axis, double AngleDegrees = 360, bool
DeleteSource = false);
```

geometry.screw Create a helical solid or thread

Parameter record: ScrewParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
profileId	yes	string
origin	yes	object
axis	yes	object
pitch	yes	number
turns	yes	number
rightHanded	optional	boolean
deleteSource	optional	boolean

```
public sealed record ScrewParameters(string ProfileId, Point3 Origin, Point3 Axis,
    double Pitch, double Turns, bool RightHanded = true, bool DeleteSource = false);
```

geometry.sweep Sweep along a path

Parameter record: SweepParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
profileId	yes	string
pathId	yes	string
deleteSources	optional	boolean

```
public sealed record SweepParameters(string ProfileId, string PathId, bool DeleteSources = false);
```

solid.boolean Boolean union subtraction or intersection

Parameter record: BooleanParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityIds	yes	array of string

JSON parameter	Required	Type
kind	yes	string

```
public sealed record BooleanParameters(string[] EntityIds, string Kind);
```

solid.chamfer Chamfer solid edges

Parameter record: ChamferParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
referenceFace	yes	object
edges	yes	array of object
distance1	yes	number
distance2	yes	number

```
public sealed record ChamferParameters(string EntityId, SubobjectRef ReferenceFace, SubobjectRef[] Edges, double
Distance1, double Distance2);
```

solid.fillet.resize Change the radius of recognized fillet faces

Parameter record: ResizeFilletParameters / worker

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object
radius	yes	number

solid.fillet Fillet solid edges

Parameter record: FilletParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
edges	yes	array of object
radius	yes	number

```
public sealed record FilletParameters(string EntityId, SubobjectRef[] Edges, double Radius);
```

solid.fillet.remove Remove fillets

Parameter record: RemoveFacesParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object

```
public sealed record RemoveFacesParameters(string EntityId, SubobjectRef[] Faces);
```

solid.primitive Create a primitive solid

Parameter record: PrimitiveParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
kind	yes	string
origin	yes	object
size	optional	object / null
axis	optional	object / null
radius	optional	number
secondRadius	optional	number

```
public sealed record PrimitiveParameters(string Kind, Point3 Origin, Point3? Size = null, Point3? Axis = null,
    double Radius = 10, double SecondRadius = 5);
```

solid.shell Hollow a solid

Parameter record: ShellParameters / worker / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityId	yes	string
openFaces	yes	array of object
thickness	yes	number

```
public sealed record ShellParameters(string EntityId,SubobjectRef[] OpenFaces,double Thickness);
```

surface.create Create a planar face

Parameter record: CreateFaceParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityId	yes	string
deleteSource	optional	boolean

```
public sealed record CreateFaceParameters(string EntityId, bool DeleteSource = false);
```

surface.sew Sew faces or shells

Parameter record: SewParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string

```
public sealed record SewParameters(string[] EntityIds);
```

surface.trim Trim a standalone face

Parameter record: SurfaceTrimParameters / worker / Free / STD / Pro

JSON parameter	Required	Type
target	yes	object
cutter	yes	object

```
public sealed record SurfaceTrimParameters(SubobjectRef Target, SubobjectRef Cutter);
```

Sheet metal — operations and parameters

sheet.bend Bend along a line

Parameter record: SheetBendParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
lineId	yes	string
fixedFace	yes	object
rules	yes	object
angleDegrees	optional	number
positionReference	optional	integer

```
public sealed record SheetBendParameters(string EntityId, string LineId, SubobjectRef FixedFace, SheetRules Rules,
    double AngleDegrees = 90, int PositionReference = 0);
```

sheet.contour Create a contour flange

Parameter record: SheetContourParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
edge	yes	object
rules	yes	object
steps	yes	array of object

```
public sealed record SheetContourParameters(string EntityId, SubobjectRef Edge, SheetRules Rules, SheetStep[] Steps);
```

sheet.corner Treat a corner

Parameter record: SheetCornerParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
faces	yes	array of object
gap	yes	number
kind	optional	integer

```
public sealed record SheetCornerParameters(string EntityId, SubobjectRef[] Faces, double Gap, int Kind = 0);
```

sheet.editBend Edit a bend

Parameter record: SheetEditBendParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
bendFace	yes	object
angleDegrees	yes	number
radius	yes	number
length	yes	number
lengthReference	optional	integer

```
public sealed record SheetEditBendParameters(string EntityId, SubobjectRef BendFace, double AngleDegrees,
    double Radius, double Length, int LengthReference = 0);
```

sheet.editForm Edit or suppress a formed feature

Parameter record: SheetEditFormParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
formId	yes	string
dimensions	yes	object
delete	optional	boolean

```
public sealed record SheetEditFormParameters(string EntityId, string FormId, SheetFormDimensions Dimensions, bool
Delete = false);
```

sheet.flange Create an edge flange

Parameter record: SheetFlangeParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
edges	yes	array of object

JSON parameter	Required	Type
rules	yes	object
length	yes	number
angleDegrees	optional	number
gap	optional	number
autoCorner	optional	boolean
positionReference	optional	integer
lengthReference	optional	integer
startDistance	optional	number / null
endDistance	optional	number / null

```
public sealed record SheetFlangeParameters(string EntityId, SubobjectRef[] Edges, SheetRules Rules,
    double Length, double AngleDegrees = 90, double Gap = .1, bool AutoCorner = true,
    int PositionReference = 0, int LengthReference = 0, double? StartDistance = null, double? EndDistance = null);
```

sheet.flatPattern Generate a flat pattern

Parameter record: SheetFlatPatternParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
placement	yes	object
bendLines	optional	boolean
annotations	optional	boolean
bendTable	optional	boolean

```
public sealed record SheetFlatPatternParameters(string EntityId, Point3 Placement, bool BendLines = true, bool
Annotations = true, bool BendTable = false);
```

sheet.form Create a formed feature

Parameter record: SheetFormParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
face	yes	object
center	yes	object

JSON parameter	Required	Type
dimensions	yes	object
rotationDegrees	optional	number
reverse	optional	boolean

```
public sealed record SheetFormParameters(string EntityId, SubobjectRef Face, Point3 Center, SheetFormDimensions
Dimensions,
    double RotationDegrees = 0, bool Reverse = false);
```

sheet.hem Create an open hem

Parameter record: SheetHemParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
edges	yes	array of object
rules	yes	object
length	yes	number
gap	yes	number

```
public sealed record SheetHemParameters(string EntityId, SubobjectRef[] Edges, SheetRules Rules, double Length, double
Gap);
```

sheet.jog Create a jog

Parameter record: SheetJogParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
edge	yes	object
rules	yes	object
height	yes	number
length	yes	number
angleDegrees	optional	number

```
public sealed record SheetJogParameters(string EntityId, SubobjectRef Edge, SheetRules Rules, double Height, double
Length, double AngleDegrees = 90);
```

sheet.loft Create or edit a rolled or lofted sheet

Parameter record: SheetLoftParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
rules	yes	object
dimensions	yes	object
entityId	optional	string / null

```
public sealed record SheetLoftParameters(SheetRules Rules, SheetLoftDimensions Dimensions, string? EntityId = null);
```

sheet.manageBend Suppress restore delete or reparent a bend

Parameter record: SheetManageBendParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
bendIndex	yes	integer
operation	yes	integer
parentIndex	optional	integer

```
public sealed record SheetManageBendParameters(string EntityId, int BendIndex, int Operation, int ParentIndex = -1);
```

sheet.plate Create a sheet-metal base plate

Parameter record: SheetPlateParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
profileId	yes	string
rules	yes	object
thicknessSide	optional	integer

```
public sealed record SheetPlateParameters(string ProfileId, SheetRules Rules, int ThicknessSide = 0);
```

sheet.recognize Recognize sheet metal

Parameter record: SheetRecognizeParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
baseFace	yes	object
rules	yes	object
detectThickness	optional	boolean

```
public sealed record SheetRecognizeParameters(string EntityId, SubobjectRef BaseFace, SheetRules Rules, bool
DetectThickness = true);
```

sheet.refold Refold sheet metal

Parameter record: SheetBodyParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string

```
public sealed record SheetBodyParameters(string EntityId);
```

sheet.relief Change bend relief

Parameter record: SheetReliefParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
edges	yes	array of object
width	yes	number
depth	yes	number
round	optional	boolean
remove	optional	boolean

```
public sealed record SheetReliefParameters(string EntityId, SubobjectRef[] Edges, double Width, double Depth, bool
Round = false, bool Remove = false);
```

sheet.rip Create a rip or slit

Parameter record: SheetRipParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
edges	yes	array of object
gap	yes	number
side	optional	integer

```
public sealed record SheetRipParameters(string EntityId, SubobjectRef[] Edges, double Gap, int Side = 2);
```

sheet.rules Change sheet-metal rules

Parameter record: SheetRulesParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string
rules	yes	object

```
public sealed record SheetRulesParameters(string EntityId, SheetRules Rules);
```

sheet.unfold Unfold sheet metal

Parameter record: SheetBodyParameters / worker / STD / Pro

Use a valid sheet-metal model with the required face/edge references. Forming unfold results represent the reference flat; they are not developed blanks from plastic-flow simulation. See Part 7 for supported selections, dimensions and limits.

JSON parameter	Required	Type
entityId	yes	string

```
public sealed record SheetBodyParameters(string EntityId);
```

Measurement and analysis — operations and parameters

analysis.mesh Create a finite element mesh

Parameter record: AnalysisMeshParameters / worker-data / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
dimension	yes	integer
meshSize	optional	number
tolerance	optional	number

```
public sealed record AnalysisMeshParameters(string[] EntityIds,int Dimension,double MeshSize=5,double Tolerance=.01);
```

analysis.assembly.contacts Find assembly contacts

Parameter record: AssemblyContactsParameters / worker-data

JSON parameter	Required	Type
entityIds	yes	array of string
tolerance	optional	number

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var selection = await cad.GetSelectionAsync(cancellationToken: ct);
string[] ids = selection.Entities.Select(e => e.Id).ToArray();
var request = doc.Operation("analysis.assembly.contacts",
    new AssemblyContactsParameters(ids), selectResult: false);
var job = await cad.StartOperationAsync(new(request, 180), ct);
while (job.State == "running")
{
    await Task.Delay(100, ct);
    job = await cad.GetJobAsync(job.Id, ct);
}
if (job.State != "completed") throw new InvalidOperationException(job.Message);
var result = await cad.GetJobDataAsync(new(job.Id, "summary"), ct);
```

analysis.assembly.mesh Mesh an assembly with materials and contacts

Parameter record: AssemblyMeshParameters / worker-data

JSON parameter	Required	Type
entityIds	yes	array of string
meshSize	optional	number
tolerance	optional	number
excludedContactGroups	optional	array / null

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var selection = await cad.GetSelectionAsync(cancellationToken: ct);
string[] ids = selection.Entities.Select(e => e.Id).ToArray();
var request = doc.Operation("analysis.assembly.mesh",
    new AssemblyMeshParameters(ids, MeshSize: 5, ExcludedContactGroups: []), selectResult: false);
var job = await cad.StartOperationAsync(new(request, 180), ct);
while (job.State == "running")
{
    await Task.Delay(100, ct);
    job = await cad.GetJobAsync(job.Id, ct);
}
if (job.State != "completed") throw new InvalidOperationException(job.Message);
var result = await cad.GetJobDataAsync(new(job.Id, "summary"), ct);
```

analysis.assembly.solve Solve assembly structural or thermal analysis

Parameter record: AssemblySolveParameters / worker-data

JSON parameter	Required	Type
mesh	yes	object
bodyMaterials	yes	array of object
conditions	yes	array of object
kind	optional	string
transientThermal	optional	boolean
initialTemperature	optional	number
endTime	optional	number
timeStep	optional	number
outputCount	optional	integer

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("analysis.assembly.solve",
    new AssemblySolveParameters(assemblyMesh,
        [new AnalysisConditionData("temperature", Target: "node", NodeIds: leftNodes, X: 0),
         new AnalysisConditionData("temperature", Target: "node", NodeIds: rightNodes, X: 100)],
        Kind: "heat"), selectResult: false);
var job = await cad.StartOperationAsync(new(request, 180), ct);
while (job.State == "running")
{
    await Task.Delay(100, ct);
    job = await cad.GetJobAsync(job.Id, ct);
}
if (job.State != "completed") throw new InvalidOperationException(job.Message);
var result = await cad.GetJobDataAsync(new(job.Id, "summary"), ct);
```

analysis.solve Solve elasticity or heat transfer

Parameter record: AnalysisSolveParameters / worker-data / STD / Pro

JSON parameter	Required	Type
mesh	yes	object
material	yes	object
conditions	yes	array of object
kind	optional	string
thickness	optional	number
planeStrain	optional	boolean
transientThermal	optional	boolean
initialTemperature	optional	number
endTime	optional	number
timeStep	optional	number
outputCount	optional	integer

```
public sealed record AnalysisSolveParameters(AnalysisMeshData Mesh,AnalysisMaterialData
Material,AnalysisConditionData[] Conditions,string Kind="elastic",double Thickness=1,bool PlaneStrain=false,
    bool TransientThermal=false,double InitialTemperature=20,double EndTime=60,double TimeStep=1,int OutputCount=61);
```

measure.contour Measure a closed contour and its section properties

Parameter record: ContourMeasureParameters / worker-data

JSON parameter	Required	Type
seedId	yes	string
candidateIds	yes	array of string
origin	yes	object
normal	yes	object
xDirection	yes	object

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var selection = await cad.GetSelectionAsync(cancellationToken: ct);
string[] ids = selection.Entities.Select(e => e.Id).ToArray();
var request = doc.Operation("measure.contour",
    new ContourMeasureParameters(ids[0], ids,
        new(0, 0, 0), new(0, 0, 1), new(1, 0, 0)), selectResult: false);
var job = await cad.StartOperationAsync(new(request, 180), ct);
while (job.State == "running")
{
    await Task.Delay(100, ct);
    job = await cad.GetJobAsync(job.Id, ct);
}
if (job.State != "completed") throw new InvalidOperationException(job.Message);
var result = await cad.GetJobDataAsync(new(job.Id, "summary"), ct);
```

measure.highlight Highlight the recognized contour

Parameter record: ContourHighlightParameters / immediate

JSON parameter	Required	Type
contour	optional	object / null
clear	optional	boolean

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var selection = await cad.GetSelectionAsync(cancellationToken: ct);
string[] ids = selection.Entities.Select(e => e.Id).ToArray();
var request = doc.Operation("measure.highlight",
    new ContourHighlightParameters(new ContourMeasureParameters(ids[0], ids,
        new(0, 0, 0), new(0, 0, 1), new(1, 0, 0))), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

measure.geometry Measure geometry and sections

Parameter record: MeasureParameters / worker-data / Free / STD / Pro

JSON parameter	Required	Type
entityIds	yes	array of string
tolerance	optional	number

```
public sealed record MeasureParameters(string[] EntityIds,double Tolerance=0.001);
```

solid.interference Check interference and contact

Parameter record: InterferenceParameters / worker-data / Free / STD / Pro

Geometric validity depends on the selected faces/edges and dimensions. Inspect the job result before committing; an unsupported combination is reported as an error and is not a successful edit. Do not assume that an arbitrary intersection, fillet or offset will succeed.

JSON parameter	Required	Type
entityIds	yes	array of string
tolerance	optional	number
pairs	optional	array / null

```
public sealed record InterferenceParameters(string[] EntityIds, double Tolerance = 0.000001, InterferencePair[]? Pairs
= null);
```

Layouts and display — operations and parameters

layout.activate Activate a layout

Parameter record: LayoutActivateParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
name	yes	string

```
public sealed record LayoutActivateParameters(string Name);
```

layout.annotations Add or replace editable text leaders

Parameter record: LayoutAnnotationsParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
name	yes	string
removeIndices	yes	array of integer
leaders	yes	array of object

```
public sealed record LayoutAnnotationsParameters(string Name,int[] RemoveIndices,LayoutLeaderParameters[] Leaders);
```

layout.view Create, edit or remove a layout view

Parameter record: LayoutViewEditParameters / immediate

JSON parameter	Required	Type
name	yes	string
mode	yes	string
index	optional	integer
view	optional	object / null

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var layouts = await cad.QueryAsync(new(doc.Id, doc.Revision, "layouts", []), ct);
string layoutName = layouts.Data[0].GetProperty("name").GetString()!;
var request = doc.Operation("layout.view",
    new LayoutViewEditParameters(layoutName, "add", View:
        new LayoutViewParameters(new(0, 0, -1), new(0, 1, 0),
            new(0, 0, 0), new(60, 60, 0), 1,
            new(10, 110, 10, 110), ["Standard"], "hiddenLines")), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

section.plane Configure the live section plane

Parameter record: SectionPlaneParameters / immediate

JSON parameter	Required	Type
origin	yes	object
normal	yes	object
xDirection	yes	object
width	optional	number
height	optional	number
live	optional	boolean
flipped	optional	boolean
active	optional	boolean
entityId	optional	string / null

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("section.plane",
    new SectionPlaneParameters(new(0, 0, 5), new(0, 0, 1), new(1, 0, 0),
        Width: 100, Height: 75, Live: true, Flipped: false), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

section.generate Generate section geometry

Parameter record: SectionGenerateParameters / worker

JSON parameter	Required	Type
planeId	yes	string
entityIds	yes	array of string
insertionPoint	yes	object

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var selection = await cad.GetSelectionAsync(cancellationToken: ct);
string[] ids = selection.Entities.Select(e => e.Id).ToArray();
var planes = await cad.QueryAsync(new(doc.Id, doc.Revision, "sectionPlanes", []), ct);
string planeId = planes.Data[0].GetProperty("entity").GetProperty("id").GetString()!;
var request = doc.Operation("section.generate",
    new SectionGenerateParameters(planeId,
        ids.Where(id => id != planeId).ToArray(), new(150, 0, 0)), selectResult: false);
var job = await cad.StartOperationAsync(new(request, 180), ct);
while (job.State == "running")
{
    await Task.Delay(100, ct);
    job = await cad.GetJobAsync(job.Id, ct);
}
if (job.State != "ready") throw new InvalidOperationException(job.Message);
doc = await cad.GetDocumentAsync(ct);
var result = await cad.CommitJobAsync(new(doc.Id, doc.Revision,
```

```
    Guid.NewGuid().ToString(), job.Id), ct);
```

animation.view Configure an animation view

Parameter record: AnimationViewParameters / immediate

JSON parameter	Required	Type
name	yes	string
direction	yes	object
up	yes	object
repeatMode	optional	string
repeatCount	optional	integer
speed	optional	number
visibleLayers	optional	array / null
mode	optional	string

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("animation.view",
        RepeatMode: "count", RepeatCount: 3, Speed: 1), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

animation.drive Configure an animation drive

Parameter record: AnimationDriveParameters / immediate

JSON parameter	Required	Type
name	yes	string
kind	yes	string
entityIds	yes	array of string
axisOrigin	optional	object / null
axisDirection	optional	object / null
curveId	optional	string / null
firstDrive	optional	string / null
secondDrive	optional	string / null
dependsOn	optional	string / null
position	optional	number
nullPosition	optional	number
tangential	optional	boolean
stretch	optional	boolean

JSON parameter	Required	Type
mode	optional	string

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var selection = await cad.GetSelectionAsync(cancellationToken: ct);
string[] ids = selection.Entities.Select(e => e.Id).ToArray();
var request = doc.Operation("animation.drive",
        AxisOrigin: new(0, 0, 0), AxisDirection: new(0, 0, 1)), selectResult: false);
var result = await cad.ExecuteOperationAsync(request, ct);
```

animation.schedule Configure an animation schedule

Parameter record: AnimationScheduleParameters / immediate

JSON parameter	Required	Type
name	yes	string
tracks	yes	array of object
mode	optional	string

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("animation.schedule",
var result = await cad.ExecuteOperationAsync(request, ct);
```

animation.control Control playback and time

Parameter record: AnimationControlParameters / immediate

JSON parameter	Required	Type
view	yes	string
action	yes	string
schedule	optional	string / null
start	optional	number
end	optional	number / null
speed	optional	number / null

C# example. api is IAddinContext.Api and ct is a CancellationToken. Supply selections, names and dimensions from the current drawing.

```
// using JDesignCAD.Api; using System.Linq;
var cad = (ICadWorkspaceApi)api;
var doc = await cad.GetDocumentAsync(ct);
var request = doc.Operation("animation.control",
var result = await cad.ExecuteOperationAsync(request, ct);
```

layout.create Create a dimensioned drawing layout

Parameter record: LayoutCreateParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
name	yes	string
width	yes	number
height	yes	number
views	yes	array of object
objects	yes	array of object
dimensions	yes	array of object

```
public sealed record LayoutCreateParameters(string Name,double Width,double Height,LayoutViewParameters[] Views,
    GeometryDefinition[] Objects,LayoutDimensionParameters[] Dimensions);
```

view.activate Activate a model or layout view

Parameter record: ActivateViewParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
kind	yes	string
name	optional	string / null
zoomToFit	optional	boolean

```
public sealed record ActivateViewParameters(string Kind, string? Name = null, bool ZoomToFit = false);
```

view.capture Render a model or layout to PNG

Parameter record: DrawingCaptureParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
filePath	yes	string
layoutName	optional	string / null
width	optional	integer
height	optional	integer
overwrite	optional	boolean

```
public sealed record DrawingCaptureParameters(string FilePath,string? LayoutName=null,int Width=1600,int
Height=1100,bool Overwrite=false);
```

Files — operations and parameters

document.open Open a native drawing

Parameter record: OpenDocumentParameters / immediate / Free / STD / Pro

JSON parameter	Required	Type
filePath	yes	string

```
public sealed record OpenDocumentParameters(string FilePath);
```

Public data record definitions

The following C# declarations are extracted from public source. Values after = are defaults, ? permits null and [] denotes arrays. Also check required fields in the operation schema. Record names correspond to method request and result types.

ActivateViewParameters

```
public sealed record ActivateViewParameters(string Kind, string? Name = null, bool ZoomToFit = false);
```

AnalysisBoundaryData

```
public sealed record AnalysisBoundaryData(int Group,int[] Nodes,int Body=0);
```

AnalysisConditionData

```
public sealed record AnalysisConditionData(string Kind,int Group=0,int Node=-1,string Target="boundary",int[]?
Groups=null,int[]? NodeIds=null,int[]? BoundaryIndices=null,double X=0,double Y=0,double Z=0);
```

AnalysisMaterialData

```
public sealed record AnalysisMaterialData(string Name="Steel",double YoungModulus=200000,double PoissonRatio=.3,double
Density=7850,double Conductivity=50,double SpecificHeat=460,double ExpansionMicro=12);
```

AnalysisMeshData

```
public sealed record AnalysisMeshData(int Dimension,double[][] Points,int[][] Elements,AnalysisBoundaryData[]
Boundaries,AnalysisBoundaryData[]? EdgeBoundaries=null,int[]? VertexNodes=null);
```

AnalysisMeshParameters

```
public sealed record AnalysisMeshParameters(string[] EntityIds,int Dimension,double MeshSize=5,double Tolerance=.01);
```

AnalysisSolveParameters

```
public sealed record AnalysisSolveParameters(AnalysisMeshData Mesh,AnalysisMaterialData
Material,AnalysisConditionData[] Conditions,string Kind="elastic",double Thickness=1,bool PlaneStrain=false,
    bool TransientThermal=false,double InitialTemperature=20,double EndTime=60,double TimeStep=1,int OutputCount=61);
```

ArcDefinition

```
public sealed record ArcDefinition(Point3 Center, Point3 Start, Point3 Normal, double SweepDegrees);
```

AttributeInfo

```
public sealed record AttributeInfo(EntityInfo Entity, CadColor? Color, string? LinePattern, string? LineWidth,
    bool LayerLocked);
```

AttributePage

```
public sealed record AttributePage(DocumentInfo Document, AttributeInfo[] Items);
```

BlockAttributesParameters

```
public sealed record BlockAttributesParameters(string EntityId,BlockValue[] Values);
```

BlockCreateParameters

```
public sealed record BlockCreateParameters(string[] EntityIds,string Name,Point3 BasePoint);
```

BlockDynamicParameters

```
public sealed record BlockDynamicParameters(string EntityId,string Key,string Value);
```

BlockInsertParameters

```
public sealed record BlockInsertParameters(string Name,Point3 Position,Point3? Scale=null,double
RotationDegrees=0,Point3? Normal=null);
```

BlockValue

```
public sealed record BlockValue(string Name,string Value);
```

BooleanParameters

```
public sealed record BooleanParameters(string[] EntityIds, string Kind);
```

Bounds3

```
public sealed record Bounds3(Point3 Minimum, Point3 Maximum);
```

CadColor

```
public sealed record CadColor(string Mode, string? Rgb = null);
```

CapabilityInfo

```
public sealed record CapabilityInfo(int Major, int Minor, string CoordinateSystem, string LengthUnit,
    OperationInfo[] Operations, string[] Limitations)
{
    public LicenseCapabilities? License { get; init; }
    public FileFormatInfo[]? FileFormats { get; init; }
    /// <summary>Input methods are separate from geometry operations; points are supplied by the user in
CAD.</summary>
    public string[]? InputMethods { get; init; }
}
```

ChamferParameters

```
public sealed record ChamferParameters(string EntityId, SubobjectRef ReferenceFace, SubobjectRef[] Edges, double
Distance1, double Distance2);
```

CircleDefinition

```
public sealed record CircleDefinition(Point3 Center, double Radius, Point3 Normal);
```

CircularArrayParameters

```
public sealed record CircularArrayParameters(string[] EntityIds, Point3 Center, Point3 Axis, double StepDegrees, int
Count);
```

CommandResult

```
public sealed record CommandResult(string Message);
```

CommitJobRequest

```
public sealed record CommitJobRequest(string DocumentId, long ExpectedRevision, string OperationId, string JobId);
```

CreateCurvesRequest

```
public sealed record CreateCurvesRequest(string DocumentId, long ExpectedRevision, string OperationId,
    PolylineDefinition[] Polylines, ArcDefinition[] Arcs, bool SelectCreated = true);
```

CreateFaceParameters

```
public sealed record CreateFaceParameters(string EntityId, bool DeleteSource = false);
```

CreateFillsParameters

```
public sealed record CreateFillsParameters(FillDefinition[] Fills);
```

CreateGeometryRequest

```
public sealed record CreateGeometryRequest(string DocumentId, long ExpectedRevision, string OperationId,
    LineDefinition[] Lines, CircleDefinition[] Circles, bool SelectCreated = true);
```

CreateObjectsParameters

```
public sealed record CreateObjectsParameters(GeometryDefinition[] Objects);
```

CurveJoinParameters

```
public sealed record CurveJoinParameters(string[] EntityIds,double Tolerance=0.000001);
```

CurveOffsetParameters

```
public sealed record CurveOffsetParameters(string EntityId,Point3 PlaneNormal,double Distance,double Tolerance=0.001);
```

CurveSplitParameters

```
public sealed record CurveSplitParameters(string EntityId,double Parameter);
```

DeleteLayerParameters

```
public sealed record DeleteLayerParameters(string Name);
```

DimensionParameters

```
public sealed record DimensionParameters(string Kind,Point3[] Points,Point3 TextPosition,Point3 Normal,Point3
Direction,double Radius=0,string? Style=null);
```

DocumentInfo

```
public sealed record DocumentInfo(string Id, string? FileName, long Revision, int ObjectCount,
    bool CanUndo, bool CanRedo, string LengthUnit = "mm", string CoordinateSystem = "WCS");
```

DocumentRequest

```
public sealed record DocumentRequest(string DocumentId, long ExpectedRevision);
```

DrawingCaptureParameters

```
public sealed record DrawingCaptureParameters(string FilePath,string? LayoutName=null,int Width=1600,int
Height=1100,bool Overwrite=false);
```

EditResult

```
public sealed record EditResult(DocumentInfo Document, EntityInfo[] Entities, bool Changed);
```

EllipseDefinition

```
public sealed record EllipseDefinition(Point3 Center, Point3 MajorAxis, Point3 MinorAxis,
    double StartDegrees = 0, double SweepDegrees = 360);
```

EntityDetail

```
public sealed record EntityDetail(EntityInfo Entity, bool LayerLocked, string GeometryStatus,
    LineDefinition? Line = null, CircleDefinition? Circle = null, ArcDefinition? Arc = null, PolylineDefinition?
Polyline = null);
```

EntityDetailsRequest

```
public sealed record EntityDetailsRequest(string DocumentId, long ExpectedRevision, string[] EntityIds);
```

EntityDetailsResult

```
public sealed record EntityDetailsResult(DocumentInfo Document, EntityDetail[] Entities);
```

EntityEditRequest

```
public sealed record EntityEditRequest(string DocumentId, long ExpectedRevision, string OperationId, string[]
EntityIds);
```

EntityInfo

```
public sealed record EntityInfo(string Id, string Kind, string? Layer, Bounds3? Bounds);
```

EntityPage

```
public sealed record EntityPage(DocumentInfo Document, EntityInfo[] Entities, int Total, int Offset);
```

EntityQuery

```
public sealed record EntityQuery(string DocumentId, long ExpectedRevision, int Offset = 0, int Limit = 256);
```

ExplodeParameters

```
public sealed record ExplodeParameters(string[] EntityIds);
```

ExportDocumentRequest

```
public sealed record ExportDocumentRequest(string DocumentId, long ExpectedRevision, string OperationId,
    string FilePath, string Format, bool Overwrite = false, string[]? EntityIds = null,
    Point3? BasePoint = null, IfcFileOptions? Ifc = null, int TimeoutSeconds = 300);
```

ExtrudeParameters

```
public sealed record ExtrudeParameters(string[] EntityIds, Point3 Direction, bool Solid = true, bool DeleteSources =
false);
```

FaceMoveParameters

```
public sealed record FaceMoveParameters(string EntityId,SubobjectRef[] Faces,Point3 Displacement,bool
MoveConnected=false);
```

FaceOffsetParameters

```
public sealed record FaceOffsetParameters(string EntityId, SubobjectRef[] Faces, double Distance);
```

FaceRadiusParameters

```
public sealed record FaceRadiusParameters(string EntityId,SubobjectRef[] Faces,double Radius);
```

FaceRotateParameters

```
public sealed record FaceRotateParameters(string EntityId,SubobjectRef[] Faces,Point3 Origin,Point3 Axis,double
AngleDegrees,bool MoveConnected=false);
```

FileFormatInfo

```
public sealed record FileFormatInfo(string Format, string[] Extensions, bool CanSave, bool CanExport,
    bool Available, string Description, string? UnavailableCode = null, LicenseRestriction? LicenseRestriction =
null);
```

FileOperationInfo

```
public sealed record FileOperationInfo(string OperationId, string DocumentId, long SourceRevision,
    string Kind, string FilePath, string Format, string State, bool DocumentSaved = false,
    long? Bytes = null, string? Sha256 = null, string[]? Warnings = null, FileOutputError? Error = null)
{
    /// <summary>OBJ's uniquely named MTL sidecar. Keep these files beside the main output when transferring
it.</summary>
    public FileOutputArtifact[] AdditionalFiles { get; init; } = [];
}
```

FileOutputArtifact

```
public sealed record FileOutputArtifact(string FilePath, long Bytes, string Sha256);
```

FileOutputError

```
public sealed record FileOutputError(string Code, string Message, bool Retryable = false,
    LicenseRestriction? LicenseRestriction = null);
```

FillAppearance

```
public sealed record FillAppearance(CadColor Color, string Pattern = "none",
    CadColor? Color2 = null, double AngleDegrees = 0, int Transparency = 0);
```

FillDefinition

```
public sealed record FillDefinition(GeometryDefinition Boundary, FillAppearance Appearance, string? Layer = null);
```

FilletParameters

```
public sealed record FilletParameters(string EntityId, SubobjectRef[] Edges, double Radius);
```

GeometryDefinition

```
public sealed record GeometryDefinition(LineDefinition? Line = null, CircleDefinition? Circle = null,
    ArcDefinition? Arc = null, PolylineDefinition? Polyline = null, EllipseDefinition? Ellipse = null,
    SplineDefinition? Spline = null, TextDefinition? Text = null, Point3? Point = null);
```

GeometryUpdate

```
public sealed record GeometryUpdate(string EntityId, GeometryDefinition Geometry);
```

HatchParameters

```
public sealed record HatchParameters(string ProfileId,string? Style=null);
```

HelixParameters

```
public sealed record HelixParameters(Point3 Center, Point3 Axis, double BaseRadius, double TopRadius,
    double Height, double Turns, int SegmentsPerTurn = 48, bool Clockwise = false, Point3? StartDirection = null);
```

HistoryRequest

```
public sealed record HistoryRequest(string DocumentId, long ExpectedRevision, string OperationId);
```

IfcFileOptions

```
public sealed record IfcFileOptions(string Output = "ifc4Advanced", string? Name = null);
```

ImportImageParameters

```
public sealed record ImportImageParameters(string FilePath, Point3 Location, Point3 WidthAxis, Point3 HeightAxis);
```

InteractiveCommandInfo

```
public sealed record InteractiveCommandInfo(string Id,string Group,bool Available,string? UnavailableReason=null)
{
    public string? UnavailableCode { get; init; }
    public LicenseRestriction? LicenseRestriction { get; init; }
```

```
}
```

InteractiveCommandRequest

```
public sealed record InteractiveCommandRequest(string DocumentId,long ExpectedRevision,string OperationId,string
CommandId);
```

InteractiveCommandResult

```
public sealed record InteractiveCommandResult(string CommandId,string State,string Message);
```

InterferencePair

```
public sealed record InterferencePair(string FirstEntityId, string SecondEntityId);
```

InterferenceParameters

```
public sealed record InterferenceParameters(string[] EntityIds, double Tolerance = 0.000001, InterferencePair[]? Pairs
= null);
```

JobDataPage

```
public sealed record JobDataPage(JobInfo Job,string Section,JsonElement Data,int Total,int Offset);
```

JobDataRequest

```
public sealed record JobDataRequest(string JobId,string Section="summary",int Offset=0,int Limit=128);
```

JobInfo

```
public sealed record JobInfo(string Id, string DocumentId, long SourceRevision, string Operation, string State,
    string Stage, int ResultCount = 0, string? ErrorCode = null, string? Message = null);
```

LayerInfo

```
public sealed record LayerInfo(string Name, CadColor Color, string? LinePattern, string? LineWidth,
    bool Locked, int Transparency, bool Current);
```

LayerPage

```
public sealed record LayerPage(DocumentInfo Document, LayerInfo[] Layers, string[] LinePatterns, string[] LineWidths);
```

LayoutActivateParameters

```
public sealed record LayoutActivateParameters(string Name);
```

LayoutAnnotationsParameters

```
public sealed record LayoutAnnotationsParameters(string Name,int[] RemoveIndices,LayoutLeaderParameters[] Leaders);
```

LayoutCreateParameters

```
public sealed record LayoutCreateParameters(string Name,double Width,double Height,LayoutViewParameters[] Views,
    GeometryDefinition[] Objects,LayoutDimensionParameters[] Dimensions);
```

LayoutDimensionParameters

```
public sealed record LayoutDimensionParameters(DimensionParameters Dimension,double ValueScale=1,double
TextHeight=2.5);
```

LayoutLeaderParameters

```
public sealed record LayoutLeaderParameters(string Text,Point3 Anchor,Point3 Landing,double TextHeight=3,
    string Font="Yu Gothic",Point3[]? Elbows=null,CadColor? Color=null);
```

LayoutRectangle

```
public sealed record LayoutRectangle(double Left,double Right,double Bottom,double Top);
```

LayoutViewParameters

```
public sealed record LayoutViewParameters(Point3 Direction,Point3 Up,Point3 ModelOrigin,Point3 PaperOrigin,
    double Scale,LayoutRectangle Clip,string[] VisibleLayers,string Display="hiddenLines");
```

LicenseCapabilities

```
public sealed record LicenseCapabilities(string CurrentPlan, bool CustomAddinsAllowed, bool McpAvailable = true,
string FreeAddinPolicy = "bundled_sample_only");
```

LicenseRestriction

```
public sealed record LicenseRestriction(string Feature, string Operation, string CurrentPlan, string[] RequiredPlans,
bool Retryable = false);
```

LineDefinition

```
public sealed record LineDefinition(Point3 Start, Point3 End);
```

LinearArrayParameters

```
public sealed record LinearArrayParameters(string[] EntityIds, Point3 Step, int Count, bool IncludeOriginalCopy =
false);
```

LoftParameters

```
public sealed record LoftParameters(string[] ProfileIds, bool Solid = true, string? GuideId = null, bool DeleteSources
= false);
```

MeasureParameters

```
public sealed record MeasureParameters(string[] EntityIds,double Tolerance=0.001);
```

MirrorParameters

```
public sealed record MirrorParameters(string[] EntityIds, Point3 PlaneOrigin, Point3 PlaneNormal, bool Copy = false);
```

OpenDocumentParameters

```
public sealed record OpenDocumentParameters(string FilePath);
```

OperationInfo

```
public sealed record OperationInfo(string Name, string Description, string ParameterType, string Execution,
    bool Available, string? UnavailableReason = null)
{
    public string? UnavailableCode { get; init; }
    public LicenseRestriction? LicenseRestriction { get; init; }
}
```

OperationRequest

```
public sealed record OperationRequest(string DocumentId, long ExpectedRevision, string OperationId,
    string Operation, JsonElement Parameters, bool SelectResult = true);
```

OperationSchema

```
public sealed record OperationSchema(OperationInfo Operation, JsonElement ParametersSchema);
```

Point3

```
public sealed record Point3(double X, double Y, double Z);
```

PolylineDefinition

```
public sealed record PolylineDefinition(Point3[] Vertices, bool Closed = false);
```

PrimitiveParameters

```
public sealed record PrimitiveParameters(string Kind, Point3 Origin, Point3? Size = null, Point3? Axis = null,
    double Radius = 10, double SecondRadius = 5);
```

QueryRequest

```
public sealed record QueryRequest(string DocumentId,long ExpectedRevision,string Kind,string[] EntityIds);
```

QueryResult

```
public sealed record QueryResult(DocumentInfo Document,JsonElement Data);
```

RemoveFacesParameters

```
public sealed record RemoveFacesParameters(string EntityId, SubobjectRef[] Faces);
```

RevolveParameters

```
public sealed record RevolveParameters(string EntityId, Point3 Origin, Point3 Axis, double AngleDegrees = 360, bool
DeleteSource = false);
```

RibbonButton

```
public sealed record RibbonButton(string Id, string Text, string Description, IAddinCommand Command,
    bool RequiresDocument = true, string? SmallIconPath = null, string? LargeIconPath = null);
```

RibbonPanel

```
public sealed record RibbonPanel(string Id, string Text, RibbonButton[] Buttons);
```

RibbonTab

```
public sealed record RibbonTab(string Id, string Text, RibbonPanel[] Panels);
```

RotateParameters

```
public sealed record RotateParameters(string[] EntityIds, Point3 Center, Point3 Axis, double AngleDegrees, bool Copy =
false);
```

SaveDocumentRequest

```
public sealed record SaveDocumentRequest(string DocumentId, long ExpectedRevision, string OperationId,
    string FilePath, string Format, bool Overwrite = false, int TimeoutSeconds = 300);
```

ScaleParameters

```
public sealed record ScaleParameters(string[] EntityIds, Point3 Center, double Factor, bool Copy = false);
```

ScrewParameters

```
public sealed record ScrewParameters(string ProfileId, Point3 Origin, Point3 Axis,
    double Pitch, double Turns, bool RightHanded = true, bool DeleteSource = false);
```

SelectionInfo

```
public sealed record SelectionInfo(DocumentInfo Document, EntityInfo[] Entities, int Total, int Offset);
```

SetAttributesParameters

```
public sealed record SetAttributesParameters(string[] EntityIds, string? Layer = null, CadColor? Color = null,
    string? LinePattern = null, string? LineWidth = null);
```

SetFillParameters

```
public sealed record SetFillParameters(string[] EntityIds, FillAppearance Appearance);
```

SetLayerParameters

```
public sealed record SetLayerParameters(string Name, CadColor? Color = null, bool? Locked = null,
    int? Transparency = null, bool MakeCurrent = false, string? LinePattern = null, string? LineWidth = null);
```

SewParameters

```
public sealed record SewParameters(string[] EntityIds);
```

SheetBendParameters

```
public sealed record SheetBendParameters(string EntityId, string LineId, SubobjectRef FixedFace, SheetRules Rules,
    double AngleDegrees = 90, int PositionReference = 0);
```

SheetBodyParameters

```
public sealed record SheetBodyParameters(string EntityId);
```

SheetContourParameters

```
public sealed record SheetContourParameters(string EntityId, SubobjectRef Edge, SheetRules Rules, SheetStep[] Steps);
```

SheetCornerParameters

```
public sealed record SheetCornerParameters(string EntityId, SubobjectRef[] Faces, double Gap, int Kind = 0);
```

SheetEditBendParameters

```
public sealed record SheetEditBendParameters(string EntityId, SubobjectRef BendFace, double AngleDegrees,
    double Radius, double Length, int LengthReference = 0);
```

SheetEditFormParameters

```
public sealed record SheetEditFormParameters(string EntityId, string FormId, SheetFormDimensions Dimensions, bool
Delete = false);
```

SheetFlangeParameters

```
public sealed record SheetFlangeParameters(string EntityId, SubobjectRef[] Edges, SheetRules Rules,
    double Length, double AngleDegrees = 90, double Gap = .1, bool AutoCorner = true,
    int PositionReference = 0, int LengthReference = 0, double? StartDistance = null, double? EndDistance = null);
```

SheetFlatPatternParameters

```
public sealed record SheetFlatPatternParameters(string EntityId, Point3 Placement, bool BendLines = true, bool
Annotations = true, bool BendTable = false);
```

SheetFormDimensions

```
public sealed record SheetFormDimensions(int Kind, double Diameter = 6, double Height = 4, double RootRadius = 1,
    double CrownRadius = 2, double Span = 10, double PilotDiameter = 4, bool Suppressed = false)
{
    // Additive property preserves the existing positional constructor and DLL ABI.
    public double CornerRadius { get; init; } = 1;
    public double BridgeHoleDiameter { get; init; }
    /// <summary>Rectangular draw flank angle to the original sheet plane, degrees. Default 90 preserves existing
forms.</summary>
    public double WallAngleDegrees { get; init; } = 90;
}
```

SheetFormParameters

```
public sealed record SheetFormParameters(string EntityId, SubobjectRef Face, Point3 Center, SheetFormDimensions
Dimensions,
    double RotationDegrees = 0, bool Reverse = false);
```

SheetHemParameters

```
public sealed record SheetHemParameters(string EntityId, SubobjectRef[] Edges, SheetRules Rules, double Length, double
Gap);
```

SheetJogParameters

```
public sealed record SheetJogParameters(string EntityId, SubobjectRef Edge, SheetRules Rules, double Height, double
Length, double AngleDegrees = 90);
```

SheetLoftDimensions

```
public sealed record SheetLoftDimensions(int Kind = 1, int FormingMode = 1, double BottomRadius = 30, double TopRadius
= 20,
    double Height = 50, double BottomWidth = 80, double BottomDepth = 60, double TopWidth = 50, double TopDepth = 40,
    double CornerRadius = 5, double TopCornerRadius = 5, double OffsetX = 0, double OffsetY = 0,
    double SeamGap = 1, int Segments = 32, int SeamMode = 0, double SeamAngle = 0, double Angle = 350,
    Point3? Origin = null);
```

SheetLoftParameters

```
public sealed record SheetLoftParameters(SheetRules Rules, SheetLoftDimensions Dimensions, string? EntityId = null);
```

SheetManageBendParameters

```
public sealed record SheetManageBendParameters(string EntityId, int BendIndex, int Operation, int ParentIndex = -1);
```

SheetPlateParameters

```
public sealed record SheetPlateParameters(string ProfileId, SheetRules Rules, int ThicknessSide = 0);
```

SheetRecognizeParameters

```
public sealed record SheetRecognizeParameters(string EntityId, SubobjectRef BaseFace, SheetRules Rules, bool
DetectThickness = true);
```

SheetReliefParameters

```
public sealed record SheetReliefParameters(string EntityId, SubobjectRef[] Edges, double Width, double Depth, bool
Round = false, bool Remove = false);
```

SheetRipParameters

```
public sealed record SheetRipParameters(string EntityId, SubobjectRef[] Edges, double Gap, int Side = 2);
```

SheetRules

```
public sealed record SheetRules(double Thickness = 1, double Radius = 1, double KFactor = .4,
    int ReliefType = 0, double ReliefWidth = 1, double ReliefDepth = 2);
```

SheetRulesParameters

```
public sealed record SheetRulesParameters(string EntityId, SheetRules Rules);
```

SheetStep

```
public sealed record SheetStep(double Length, double AngleDegrees, double Radius);
```

ShellParameters

```
public sealed record ShellParameters(string EntityId,SubobjectRef[] OpenFaces,double Thickness);
```

SplineDefinition

```
public sealed record SplineDefinition(Point3[] ThroughPoints, int Degree = 3, bool Closed = false);
```

StartOperationRequest

```
public sealed record StartOperationRequest(OperationRequest Request, int TimeoutSeconds = 120);
```

SubobjectRef

```
public sealed record SubobjectRef(string EntityId, long Revision, string Kind, int Index);
```

SurfaceLabelParameters

```
public sealed record SurfaceLabelParameters(SubobjectRef Face, Point3 Center, double Width=30, double Height=10,
    double RotationDegrees=0, string? Text=null, string? ImagePath=null, string FontName="Yu Gothic",
    string Color="000000", string Mode="add", string? LabelId=null);
```

SurfaceTrimParameters

```
public sealed record SurfaceTrimParameters(SubobjectRef Target, SubobjectRef Cutter);
```

SweepParameters

```
public sealed record SweepParameters(string ProfileId, string PathId, bool DeleteSources = false);
```

TextDefinition

```
public sealed record TextDefinition(string Content, Point3 Location, Point3 XDirection, Point3 YDirection,
    double Height, string Font = "Arial");
```

TextUpdate

```
public sealed record TextUpdate(string EntityId, string Content);
```

TopologyItem

```
public sealed record TopologyItem(SubobjectRef Reference, string GeometryType, Bounds3? Bounds,
    Point3? Position = null, Point3? Normal = null, double? Length = null);
```

TopologyPage

```
public sealed record TopologyPage(DocumentInfo Document, TopologyItem[] Items, int Total, int Offset);
```

TopologyRequest

```
public sealed record TopologyRequest(string DocumentId, long ExpectedRevision, string EntityId,
    string Kind = "face", int Offset = 0, int Limit = 128);
```

TranslateEntitiesRequest

```
public sealed record TranslateEntitiesRequest(string DocumentId, long ExpectedRevision, string OperationId,
    string[] EntityIds, Point3 Displacement, bool Copy = false, bool SelectResult = true);
```

TrimCurveParameters

```
public sealed record TrimCurveParameters(string EntityId, double StartParameter, double EndParameter);
```

UpdateGeometryParameters

```
public sealed record UpdateGeometryParameters(GeometryUpdate[] Updates);
```

UpdateTextParameters

```
public sealed record UpdateTextParameters(TextUpdate[] Updates);
```

AnalysisContactData

```
public sealed record AnalysisContactData(int Group,int Body1,int Body2,double Area,bool Enabled=true,double[][]?
Outline=null);
```

AnimationControlParameters

```
public sealed record AnimationControlParameters(string View, string Action, string? Schedule=null,
    double Start=0, double? End=null, double? Speed=null);
```

AnimationDriveParameters

```
public sealed record AnimationDriveParameters(string Name, string Kind, string[] EntityIds,
    Point3? AxisOrigin=null, Point3? AxisDirection=null, string? CurveId=null,
    string? FirstDrive=null, string? SecondDrive=null, string? DependsOn=null,
    double Position=0, double NullPosition=0, bool Tangential=true, bool Stretch=false, string Mode="upsert");
```

AnimationKeyframe

```
public sealed record AnimationKeyframe(double Time, double Position, bool ToCurveEnd=false);
```

AnimationScheduleParameters

```
public sealed record AnimationScheduleParameters(string Name, AnimationTrack[] Tracks, string Mode="upsert");
```

AnimationTrack

```
public sealed record AnimationTrack(string Drive, AnimationKeyframe[] Keys);
```

AnimationViewParameters

```
public sealed record AnimationViewParameters(string Name, Point3 Direction, Point3 Up,
    string RepeatMode="once", int RepeatCount=1, double Speed=1, string[]? VisibleLayers=null, string Mode="upsert");
```

AssemblyContactsParameters

```
public sealed record AssemblyContactsParameters(string[] EntityIds, double Tolerance=.01);
```

AssemblyMeshData

```
public sealed record AssemblyMeshData(AnalysisMeshData Mesh,int[] ElementBodies,string[] BodyNames,
    AnalysisContactData[] Contacts,AnalysisBoundaryData[] BodySurfaces);
```

AssemblyMeshParameters

```
public sealed record AssemblyMeshParameters(string[] EntityIds, double MeshSize=5, double Tolerance=.01, int[]?
ExcludedContactGroups=null);
```

AssemblySolveParameters

```
public sealed record AssemblySolveParameters(AssemblyMeshData Mesh,AnalysisMaterialData[] BodyMaterials,
    AnalysisConditionData[] Conditions,string Kind="elastic",bool TransientThermal=false,
    double InitialTemperature=20,double EndTime=60,double TimeStep=1,int OutputCount=61);
```

CadInputRequest

```
public sealed record CadInputRequest(string DocumentId, long ExpectedRevision, string OperationId,
    string Kind, string Prompt, PointInputOptions? PointOptions = null,
    SelectionInputOptions? SelectionOptions = null, ValueInputOptions? ValueOptions = null, bool AllowNone = false);
```

CadInputResult

```
public sealed record CadInputResult(string InputId, string Kind, string State, DocumentInfo Document,
    Point3? Point = null, double? Number = null, string? Text = null, EntityInfo[]? Entities = null,
    Point3? PickPoint = null, Point3? CursorPoint = null, string? Reason = null)
{
    public bool Accepted => State == "accepted";
    /// <summary>Face/edge picks from a top-level body, with the same index/revision convention as GetTopologyAsync.
    /// Entities contains the owning bodies. Subobject selection uses clicks, not window/preselection.</summary>
    public SubobjectRef[]? Subobjects { get; init; }
}
```

ClothoidBasicParameters

```
public sealed record ClothoidBasicParameters(Point3 Entry, Point3 Intersection, Point3 Exit,
    Point3 Normal, double Parameter, double Radius);
```

ClothoidParameters

```
public sealed record ClothoidParameters(Point3 First, Point3 Direction, Point3 Normal,
    double Parameter, double StartLength, double EndLength, bool Clockwise = false, bool ChordDirection = false,
string? EntityId = null);
```

ClothoidStationParameters

```
public sealed record ClothoidStationParameters(string EntityId, double Start, double Left, double Right,
    Point3 Normal, double? Interval = null, bool Tangent = false);
```

ContourHighlightParameters

```
public sealed record ContourHighlightParameters(ContourMeasureParameters? Contour = null, bool Clear = false);
```

ContourMeasureParameters

```
public sealed record ContourMeasureParameters(string SeedId, string[] CandidateIds,
    Point3 Origin, Point3 Normal, Point3 XDirection);
```

InputPreviewRequest

```
public sealed record InputPreviewRequest(string InputId, LineDefinition[] Lines, CircleDefinition[] Circles);
```

LayoutViewEditParameters

```
public sealed record LayoutViewEditParameters(string Name, string Mode, int Index = -1, LayoutViewParameters? View =
null);
```

LinePatternParameters

```
public sealed record LinePatternParameters(string Name, double[] Pattern, bool MakeCurrent = false, string Scale =
"layout");
```

LineWidthParameters

```
public sealed record LineWidthParameters(string Name, double Width, bool MakeCurrent = false, string Scale =
"layout");
```

PointInputOptions

```
public sealed record PointInputOptions(Point3? BasePoint = null, bool RubberBand = true);
```

ResizeFilletParameters

```
public sealed record ResizeFilletParameters(string EntityId, SubobjectRef[] Faces, double Radius);
```

SectionGenerateParameters

```
public sealed record SectionGenerateParameters(string PlaneId, string[] EntityIds, Point3 InsertionPoint);
```

SectionPlaneParameters

```
public sealed record SectionPlaneParameters(Point3 Origin, Point3 Normal, Point3 XDirection,
    double Width = 100, double Height = 75, bool Live = false, bool Flipped = false, bool Active = true, string?
EntityId = null);
```

SelectionFilter

```
public sealed record SelectionFilter(string[]? Kinds = null, string[]? Layers = null, bool IncludeLocked = false);
```

SelectionInputOptions

```
public sealed record SelectionInputOptions(SelectionFilter? Filter = null, bool UsePreselection = false,
    int MinimumCount = 1, int MaximumCount = 1000, string Target = "entity");
```

SelectionRegionRequest

```
public sealed record SelectionRegionRequest(string DocumentId, long ExpectedRevision, string Mode,
    Point3? First = null, Point3? Second = null, SelectionFilter? Filter = null, bool SetSelection = true);
```

ValueInputOptions

```
public sealed record ValueInputOptions(string? DefaultText = null, double? DefaultNumber = null,
    double? Minimum = null, double? Maximum = null, string[]? Keywords = null, bool AllowSpaces = true);
```