A VISUAL REFERENCE · pic

pic,
one change at a time

Complete source. One deliberate change. Its actual result. Follow an archive record through inspection, measurement and publication. Named geometric objects make positions explicit; short programs keep the drawings editable.

The Org file is canonical. Read the searchable HTML beside your editor or use the PDF offline. All measurements and operational quantities are synthetic teaching fixtures.

> **Last changes: 2026-11-16.** Original examples, pinned native rendering and portable source/configuration downloads.

72
worked examples

12
topic families

1
canonical Org source

All diagrams at a glance

Choose a diagram to open its complete pic source, explanation and downloads.

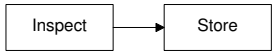
01 · First diagrams

6 diagrams

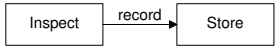
Begin with one action, then make the relation and reading direction explicit.



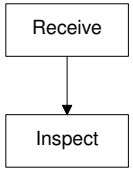
PIC-01-001
Bound an inspection



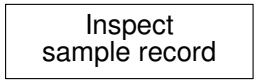
PIC-01-002
Connect the next action



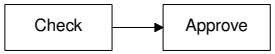
PIC-01-003
Name the handoff



PIC-01-004
Stack a process vertically



PIC-01-005
Split a long label



PIC-01-006
Explain the source

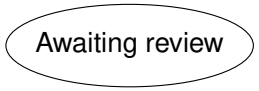
02 · Shapes and visual roles

6 diagrams

Choose a geometric primitive that helps distinguish an action, state or path.



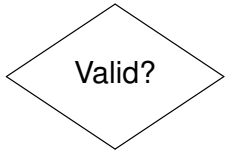
PIC-02-001
Mark a state with a circle



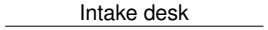
PIC-02-002
Stretch a state into an ellipse



PIC-02-003
Round an activity boundary



PIC-02-004
Construct a decision diamond



PIC-02-005
Separate a boundary from a connection

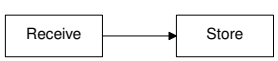


PIC-02-006
Bend a relation smoothly

03 · Anchors and alignment

6 diagrams

Use names and attachment points to express relationships between positions.



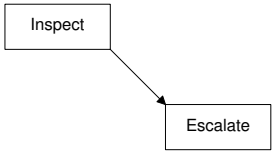
PIC-03-001

Name the objects



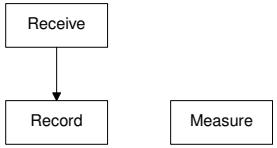
PIC-03-002

Align an edge to an anchor



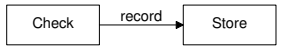
PIC-03-003

Offset a related activity



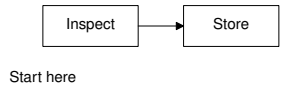
PIC-03-004

Share a horizontal coordinate



PIC-03-005

Place a note between activities



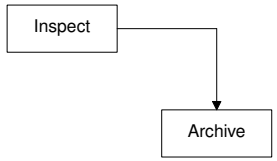
PIC-03-006

Save the current position

04 · Connections and routes

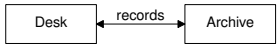
6 diagrams

Choose the route and arrow direction deliberately; geometry does not define process semantics.



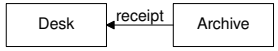
PIC-04-001

Route through an elbow



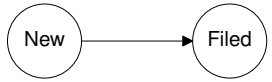
PIC-04-002

Show a two-way exchange



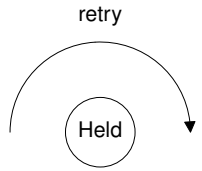
PIC-04-003

Reverse the arrowhead



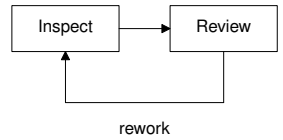
PIC-04-004

Stop at circular boundaries



PIC-04-005

Arc around a state



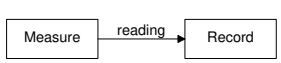
PIC-04-006

Route a rework loop below

05 · Labels and typography

6 diagrams

Let labels explain the relation without hiding its geometry.



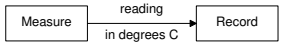
PIC-05-001

Place a label above a line



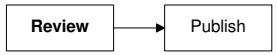
PIC-05-002

Align a note to the left



PIC-05-003

Label both sides of a connector



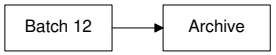
PIC-05-004

Emphasize one word



PIC-05-005

Change the label size locally



PIC-05-006

Format a computed value

06 · Appearance and emphasis

6 diagrams

Use fill, stroke and line style as redundant cues alongside the text.



PIC-06-001

Fill an activity lightly



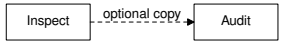
PIC-06-002

Color an activity outline



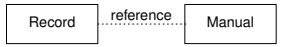
PIC-06-003

Thicken an important boundary



PIC-06-004

Dash an optional relation



PIC-06-005

Dot a supporting relation



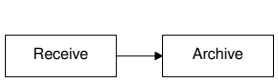
PIC-06-006

Keep scaffolding invisible

07 · Variables and expressions

6 diagrams

Make dimensions follow named values so a deliberate edit propagates through the diagram.



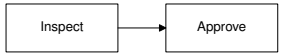
PIC-07-001

Name a shared width



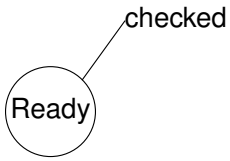
PIC-07-002

Derive the row spacing



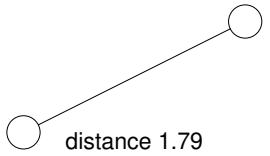
PIC-07-003

Read an object dimension



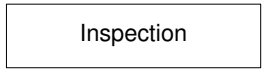
PIC-07-004

Place a radial note



PIC-07-005

Calculate a diagonal distance



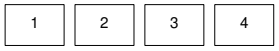
PIC-07-006

Select a width conditionally

08 · Repetition and macros

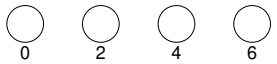
6 diagrams

Repeat a geometric idea with explicit parameters instead of duplicating unrelated edits.



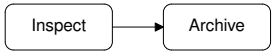
PIC-08-001

Repeat a row with a loop



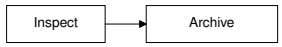
PIC-08-002

Choose the loop step



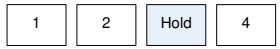
PIC-08-003

Define an activity macro



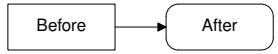
PIC-08-004

Pass a geometric parameter



PIC-08-005

Highlight one loop iteration



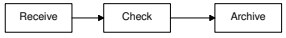
PIC-08-006

Replace a macro definition

09 · Blocks and components

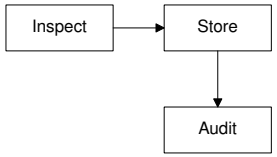
6 diagrams

Group related geometry and expose only the attachment points a surrounding diagram needs.



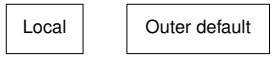
PIC-09-001

Group an intake component



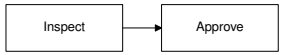
PIC-09-002

Attach to an internal port



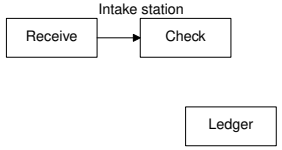
PIC-09-003

Keep a variable local



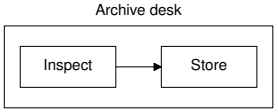
PIC-09-004

Copy the preceding shape size



PIC-09-005

Nest a reusable subdiagram



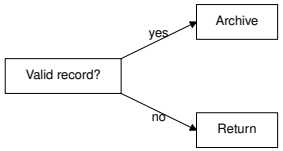
PIC-09-006

Frame a composite object

10 · Process patterns

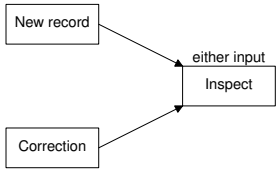
6 diagrams

State whether arrows mean ordering, alternatives or required parallel work; pic draws the supplied convention.



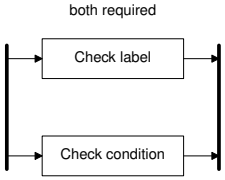
PIC-10-001

Branch on a check



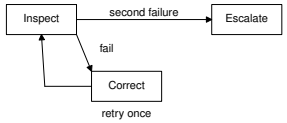
PIC-10-002

Merge alternative paths



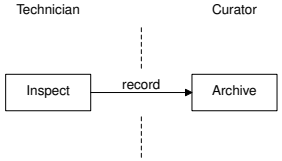
PIC-10-003

Require both parallel checks



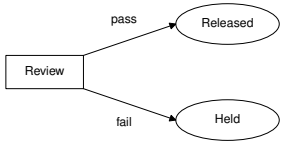
PIC-10-004

Bound a retry route



PIC-10-005

Show an ownership handoff



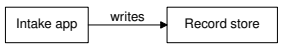
PIC-10-006

Separate failure from completion

11 · Technical relationships

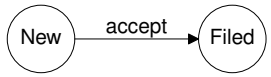
6 diagrams

Reuse the same geometry while naming whether a connector means a call, dependency, transition or pointer.



PIC-11-001

Map a component call



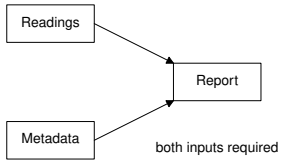
PIC-11-002

Draw a state transition



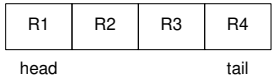
PIC-11-003

Trace a file lifecycle



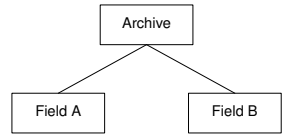
PIC-11-004

Map a dependency fan-in



PIC-11-005

Lay out queue slots



PIC-11-006

Arrange a small hierarchy

12 · Finished diagrams

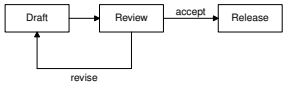
6 diagrams

Combine established primitives into small original diagrams with an explicit communication goal.



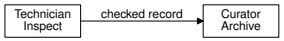
PIC-12-001

Finish an intake overview



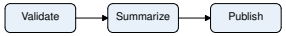
PIC-12-002

Finish a review feedback diagram



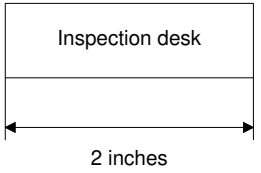
PIC-12-003

Finish a record handoff



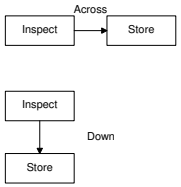
PIC-12-004

Finish a publication pipeline



PIC-12-005

Add a dimension annotation



PIC-12-006

Compare direction within one composition

READING THE TOUR

Three ways to see a change

Read the source

Each example is a complete file. Identify the syntax introduced before changing it.

Inspect the output

Open a gallery thumbnail, compare the native figure and follow a sibling example.

Make a change

Download the source/configuration ZIP, make the suggested edit and repeat its exact command.

Browse this tour with the gallery, search and topic filters. Stable IDs identify chapters and examples; Back to diagram overview returns to the corresponding thumbnail. The shared configuration sets appearance, never example data.

Reproduce the diagrams

The reviewed native toolchain is **GNU pic and groff 1.25.0, pdfcrop 1.42, Ghostscript 10.08.0 and Poppler 26.11.0**. Exact checksums and publisher versions are recorded in `VERSIONS.json`. Install GNU groff, Ghostscript, Poppler and TeX Live with pdfcrop. The driver checks the native versions and the bundled Nimbus Sans font programs. Native PDF is produced by groff with all used fonts embedded, then cropped by pdfcrop; Poppler creates the SVG and PNG companions.

Install the shared publisher with `make setup` in its sibling directory. Activate that Python environment before using a standalone bundle, or install its pinned `requirements.txt` in a fresh environment. The published HTML embeds figures and works offline; keep the companion directories for individual downloads.

```
make full
make check
tools/render --version
tools/render examples/PIC-01-001.pic
```

Command or configuration	Meaning in this tour
`tools/render SOURCE`	Run the pinned native pipeline and write PDF, SVG and PNG.
`config/render.json`	Reviewed tool versions and bundled font checksums.
`SOURCE_DATE_EPOCH=0`	Fix native build-date inputs; the driver removes PDF date metadata.
`make full`	Explicitly execute reviewed programs, then build HTML and PDF.
`make`	Reuse verified local results; never execute examples implicitly.
`make check`	Verify output hashes, example IDs and offline links.
`pic -S`	Expand the complete drawing with native safe mode.
`groff -Tpdf -P-e -P-W -P-pa3`	Embed all used fonts, make warnings fatal and use a generous A3 working canvas.
`pdfcrop --hires --margins 10`	Trim the native PDF and preserve a ten-point margin.
`pdftocairo -svg`	Convert the native PDF to its self-contained SVG companion.
`pdftocairo -png -singlefile -r 120`	Produce the downloadable PNG at a fixed resolution.

Download the complete reproduction bundle to run all examples, or inspect the shared render configuration. After installing system tools and the shared Python dependencies, preparation and rendering use no network. SVG is self-contained vector artwork; PDF remains vector artwork in the guide. Repair sources rather than editing generated figures.

01 · First diagrams

Begin with one action, then make the relation and reading direction explicit.

PIC-01-001

Bound an inspection

A box gives an action a clear boundary. Use a verb label when the diagram describes work performed on an archive record.

Compare with groff document and diagram tour.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Inspect"
.PE
```

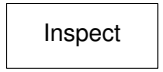
Parameter	Meaning in this example
box	Draw one rectangular activity.
"Inspect"	Place the label inside its boundary.

Render this diagram

```
tools/render examples/PIC-01-001.pic
```

Try changing: Replace Inspect with Register.

Actual output



PIC-01-001 · pic native output

pic · exit 0

PIC-01-002

Connect the next action

An arrow orders inspection before storage. Use the current drawing position for a short, unbranched process.

Compare with PIC-01-001 · Bound an inspection.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Inspect"
arrow
box "Store"
.PE
```

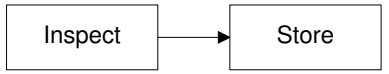
Parameter	Meaning in this example
`arrow`	Advance to the next activity and point towards it.

Render this diagram

```
tools/render examples/PIC-01-002.pic
```

Try changing: Change Store to Archive.

Actual output



PIC-01-002 · pic native output

pic · exit 0

PIC-01-003

Name the handoff

A connector label identifies what moves between activities. Use a noun here because the boxes already name the actions.

Compare with PIC-01-002 · Connect the next action.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Inspect"
arrow right .8 "record" above
box "Store"
.PE
```

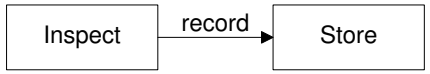
Parameter	Meaning in this example
above	Place the handoff label above the arrow.

Render this diagram

```
tools/render examples/PIC-01-003.pic
```

Try changing: Change record to checked record.

Actual output



PIC-01-003 · pic native output

pic · exit 0

PIC-01-004

Stack a process vertically

Changing direction preserves ordering while reducing width. Use a downward flow for a narrow report column.

Compare with PIC-01-003 · Name the handoff.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
down
box "Receive"
arrow
box "Inspect"
.PE
```

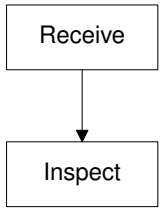
Parameter	Meaning in this example
down	Place subsequent objects below the current position.

Render this diagram

```
tools/render examples/PIC-01-004.pic
```

Try changing: Change down to right.

Actual output



PIC-01-004 · pic native output

pic · exit 0

PIC-01-005

Split a long label

Multiple strings make a label into a small stack. Use them when one activity needs a short qualifier without widening its box.

Compare with PIC-01-004 · Stack a process vertically.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box wid 1.5 "Inspect" "sample record"
.PE
```

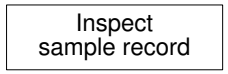
Parameter	Meaning in this example
`Two quoted strings`	Stack two lines inside the same box.

Render this diagram

```
tools/render examples/PIC-01-005.pic
```

Try changing: Replace sample record with field notes.

Actual output



PIC-01-005 · pic native output

pic · exit 0

PIC-01-006

Explain the source

A comment records the intent behind the geometry. Use it for conventions that the drawing itself cannot establish.

Compare with PIC-01-005 · Split a long label.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
# The arrow means ordering, not elapsed time.
box "Check"
arrow
box "Approve"
.PE
```

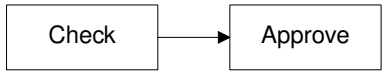
Parameter	Meaning in this example
#	Start a comment that pic does not draw.

Render this diagram

```
tools/render examples/PIC-01-006.pic
```

Try changing: Change the comment, then confirm that the figure is unchanged.

Actual output



PIC-01-006 · pic native output

pic · exit 0

02 · Shapes and visual roles

Choose a geometric primitive that helps distinguish an action, state or path.

PIC-02-001

Mark a state with a circle

A circle separates a state from an activity box. Use it for a compact condition such as a record waiting for inspection.

Compare with PIC-01-001 · Bound an inspection.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
circle "Ready"
.PE
```

Parameter	Meaning in this example
`circle`	Create a circular boundary.

Render this diagram

```
tools/render examples/PIC-02-001.pic
```

Try changing: Change Ready to Held.

Actual output



PIC-02-001 · pic native output

pic · exit 0

PIC-02-002

Stretch a state into an ellipse

An ellipse gives a state label more horizontal room. Use explicit dimensions when text length should not determine the geometry.

Compare with PIC-02-001 · Mark a state with a circle.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
ellipse wid 1.5 ht .5 "Awaiting review"
.PE
```

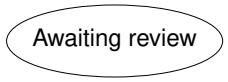
Parameter	Meaning in this example
`ellipse`	Draw an oval state boundary.
`wid / ht`	Set width and height in inches.

Render this diagram

```
tools/render examples/PIC-02-002.pic
```

Try changing: Increase wid to 1.8.

Actual output



PIC-02-002 · pic native output

pic · exit 0

PIC-02-003

Round an activity boundary

A corner radius softens a rectangular activity without changing its anchors. Use the same radius across related actions.

Compare with PIC-02-002 · Stretch a state into an ellipse.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box rad .12 "Register"
.PE
```

Parameter	Meaning in this example
`rad .12`	Round each corner with a 0.12-inch radius.

Render this diagram

```
tools/render examples/PIC-02-003.pic
```

Try changing: Change rad to .04.

Actual output



PIC-02-003 · pic native output

pic · exit 0

PIC-02-004

Construct a decision diamond

A closed polyline makes a custom boundary. Use a short question inside it and describe the outgoing alternatives when adding connections.

Compare with PIC-02-003 · Round an activity boundary.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
line from (0,.4) to (.6,0) to (0,-.4) to (-.6,0) to (0,.4)
"Valid?" at (0,0)
.PE
```

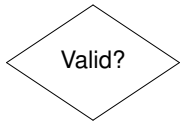
Parameter	Meaning in this example
`to`	Continue the same polyline through another vertex.
`at (0,0)`	Place the decision question independently of the path.

Render this diagram

```
tools/render examples/PIC-02-004.pic
```

Try changing: Widen both horizontal corners to .75.

Actual output



PIC-02-004 · pic native output

pic · exit 0

PIC-02-005

Separate a boundary from a connection

A plain line can delimit a region without implying direction. Use it when ownership or separation matters more than a flow.

Compare with PIC-02-004 · Construct a decision diamond.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
line right 2
"Intake desk" above at last line.c
.PE
```

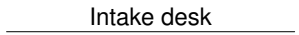
Parameter	Meaning in this example
`line`	Draw an undecorated segment.
`last line.c`	Refer to its midpoint.

Render this diagram

```
tools/render examples/PIC-02-005.pic
```

Try changing: Lengthen the boundary to 2.5.

Actual output



PIC-02-005 · pic native output

pic · exit 0

PIC-02-006

Bend a relation smoothly

A spline follows supplied waypoints with a smooth curve. Use it for a secondary relationship that should avoid a rigid elbow.

Compare with PIC-02-005 · Separate a boundary from a connection.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
spline right .7 up .4 then right .7 down .4
"Transfer route" below at last spline.c
.PE
```

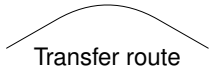
Parameter	Meaning in this example
`spline`	Interpolate a smooth path through relative movements.
`then`	Add a further waypoint.

Render this diagram

```
tools/render examples/PIC-02-006.pic
```

Try changing: Raise the middle waypoint by .2.

Actual output



PIC-02-006 · pic native output

pic · exit 0

03 · Anchors and alignment

Use names and attachment points to express relationships between positions.

PIC-03-001

Name the objects

A name makes an object available after later drawing commands. Use names for important activities rather than counting their order.

Compare with PIC-01-002 · Connect the next action.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Receive: box "Receive"
move right .8
Store: box "Store"
arrow from Receive.e to Store.w
.PE
```

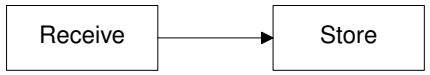
Parameter	Meaning in this example
`Name:`	Bind an uppercase name to an object.
`.e / .w`	Attach at east and west boundary points.

Render this diagram

```
tools/render examples/PIC-03-001.pic
```

Try changing: Rename Receive consistently to Intake.

Actual output



PIC-03-001 · pic native output

pic · exit 0

PIC-03-002

Align an edge to an anchor

A with clause fixes a chosen feature instead of centering the new object. Use it when boxes must share a particular boundary or port.

Compare with PIC-03-001 · Name the objects.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect"
B: box "Archive" with .w at A.e + (.7,0)
arrow from A.e to B.w
.PE
```

Parameter	Meaning in this example
`with .w at`	Place the new west boundary at a named position.

Render this diagram

```
tools/render examples/PIC-03-002.pic
```

Try changing: Change .7 to 1.

Actual output



PIC-03-002 · pic native output

pic · exit 0

PIC-03-003

Offset a related activity

A vector offset expresses displacement from an existing object. Use it for a branch whose spacing should follow its parent.

Compare with PIC-03-002 · Align an edge to an anchor.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect"
B: box "Escalate" at A + (1.6,-1)
arrow from A.se to B.nw
.PE
```

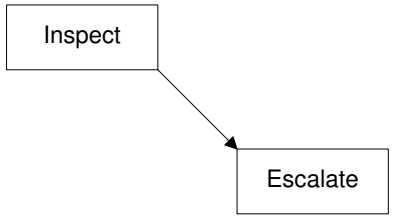
Parameter	Meaning in this example
`+(1.6,-1)`	Move right 1.6 inches and down one inch.

Render this diagram

```
tools/render examples/PIC-03-003.pic
```

Try changing: Change the vertical offset to -.7.

Actual output



PIC-03-003 · pic native output

pic · exit 0

PIC-03-004

Share a horizontal coordinate

A coordinate pair can take x and y from different named places. Use it to align an activity with a column and a row at once.

Compare with PIC-03-003 · Offset a related activity.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Receive" at (0,0)
B: box "Measure" at (1.7,-1)
C: box "Record" at (A.x,B.y)
arrow from A.s to C.n
.PE
```

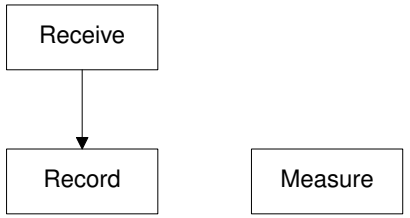
Parameter	Meaning in this example
`A.x / B.y`	Combine the first object's column with the second object's row.

Render this diagram

```
tools/render examples/PIC-03-004.pic
```

Try changing: Move B to (2,-1.3).

Actual output



PIC-03-004 · pic native output

pic · exit 0

PIC-03-005

Place a note between activities

Interpolation finds a position between two anchors. Use a fractional distance to place a shared annotation without guessing coordinates.

Compare with PIC-03-004 · Share a horizontal coordinate.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Check"
B: box "Store" at A + (2,0)
arrow from A.e to B.w
"record" above at .5 between A.e and B.w
.PE
```

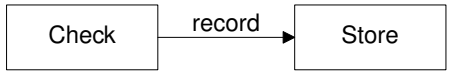
Parameter	Meaning in this example
`.5 between`	Use the midpoint of two positions.

Render this diagram

```
tools/render examples/PIC-03-005.pic
```

Try changing: Change .5 to .65.

Actual output



PIC-03-005 · pic native output

pic · exit 0

PIC-03-006

Save the current position

Here records the current drawing position. Use a named position when later objects need to return to the same construction point.

Compare with PIC-03-005 · Place a note between activities.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Start: Here
box "Inspect"
arrow
box "Store"
"Start here" below at Start + (0,-.5)
.PE
```

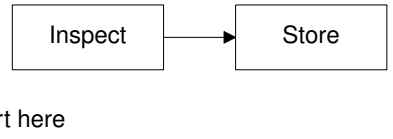
Parameter	Meaning in this example
`Here`	Capture the current location without drawing an object.

Render this diagram

```
tools/render examples/PIC-03-006.pic
```

Try changing: Move the annotation down another .2.

Actual output



PIC-03-006 · pic native output

pic · exit 0

04 · Connections and routes

Choose the route and arrow direction deliberately; geometry does not define process semantics.

PIC-04-001

Route through an elbow

A polyline can connect separated activities through orthogonal segments. Use an elbow to keep a connector away from nearby text.

Compare with PIC-03-003 · Offset a related activity.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect"
B: box "Archive" at (2,-1)
arrow from A.e to (B.x,A.y) then to B.n
.PE
```

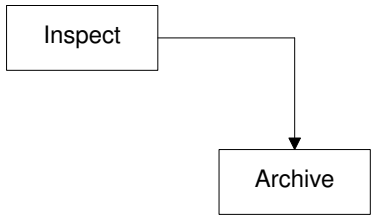
Parameter	Meaning in this example
`then to`	Continue one arrow along another segment.

Render this diagram

```
tools/render examples/PIC-04-001.pic
```

Try changing: Move Archive down to -1.4.

Actual output



PIC-04-001 · pic native output

pic · exit 0

PIC-04-002

Show a two-way exchange

Arrowheads at both ends describe a bidirectional relation. Use a noun label rather than treating it as two separate sequential actions.

Compare with PIC-04-001 · Route through an elbow.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Desk"
B: box "Archive" at (2.6,0)
line <-> from A.e to B.w "records" above
.PE
```

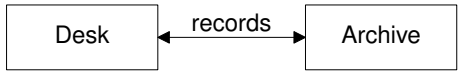
Parameter	Meaning in this example
<->	Put an arrowhead at each end.

Render this diagram

```
tools/render examples/PIC-04-002.pic
```

Try changing: Change records to status.

Actual output



PIC-04-002 · pic native output

pic · exit 0

PIC-04-003

Reverse the arrowhead

A head at the start points against the drawing motion. Use it to keep source geometry readable while representing a return relation.

Compare with PIC-04-002 · Show a two-way exchange.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Desk"
line <- right .7 "receipt" above
box "Archive"
.PE
```

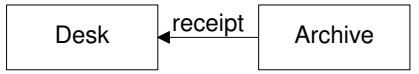
Parameter	Meaning in this example
<-	Place the arrowhead at the start of the segment.

Render this diagram

```
tools/render examples/PIC-04-003.pic
```

Try changing: Change <- to -> and explain the changed meaning.

Actual output



PIC-04-003 · pic native output

pic · exit 0

PIC-04-004

Stop at circular boundaries

Chop trims a center-to-center connection. Use explicit trimming distances when nodes are circular and their radii are known.

Compare with PIC-04-003 · Reverse the arrowhead.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: circle rad .3 "New"
B: circle rad .3 "Filed" at (1.8,0)
arrow from A to B chop .3
.PE
```

Parameter	Meaning in this example
`chop .3`	Remove 0.3 inches from both ends.

Render this diagram

```
tools/render examples/PIC-04-004.pic
```

Try changing: Increase both radii and chop to .4.

Actual output



PIC-04-004 · pic native output

pic · exit 0

PIC-04-005

Arc around a state

An arc bends around its named center. Use clockwise orientation when the relation must pass on a chosen side of the diagram.

Compare with PIC-04-004 · Stop at circular boundaries.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: circle "Held" at (0,0)
arc -> cw from (-.7,0) to (.7,0) with .c at A
"retry" at (0,.9)
.PE
```

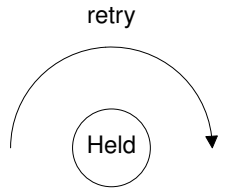
Parameter	Meaning in this example
arc	Draw a circular arc.
cw	Choose clockwise travel.

Render this diagram

```
tools/render examples/PIC-04-005.pic
```

Try changing: Change cw to ccw and move retry below.

Actual output



PIC-04-005 · pic native output

pic · exit 0

PIC-04-006

Route a rework loop below

A multi-segment return can stay outside the forward flow. Use an explicit rework label so the back edge is not mistaken for another normal handoff.

Compare with PIC-04-005 · Arc around a state.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect"
arrow
B: box "Review"
arrow from B.s down .5 then left B.x-A.x then to A.s
"rework" at ((A.x+B.x)/2,-1)
.PE
```

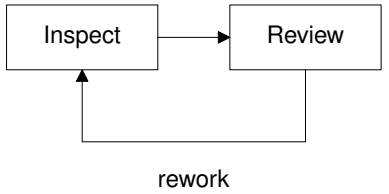
Parameter	Meaning in this example
`down / then left`	Keep a return route below the activities.

Render this diagram

```
tools/render examples/PIC-04-006.pic
```

Try changing: Move the return line down to .7.

Actual output



PIC-04-006 · pic native output

pic · exit 0

05 · Labels and typography

Let labels explain the relation without hiding its geometry.

PIC-05-001

Place a label above a line

An above label is attached to the line itself. Use it when later moves should preserve the association between text and connector.

Compare with PIC-01-003 · Name the handoff.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Measure"
arrow right 1 "reading" above
box "Record"
.PE
```

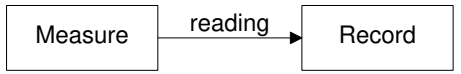
Parameter	Meaning in this example
above	Offset a label above its object.

Render this diagram

```
tools/render examples/PIC-05-001.pic
```

Try changing: Move reading below the arrow.

Actual output



PIC-05-001 · pic native output

pic · exit 0

PIC-05-002

Align a note to the left

Left justification makes a note start at its anchor. Use it for a caption next to a boundary rather than centered on it.

Compare with PIC-05-001 · Place a label above a line.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect"
"Owner: curator" ljust at A.e + (.2,0)
.PE
```

Parameter	Meaning in this example
`ljust`	Start the text at the supplied position.

Render this diagram

```
tools/render examples/PIC-05-002.pic
```

Try changing: Replace curator with technician.

Actual output



PIC-05-002 · pic native output

pic · exit 0

PIC-05-003

Label both sides of a connector

Separate labels can carry a main term and a qualifier. Use opposite sides of the connector so neither string crosses the line or obscures the other.

Compare with PIC-05-002 · Align a note to the left.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Measure"
arrow right 1.4 "reading" above "in degrees C" below
box "Record"
.PE
```

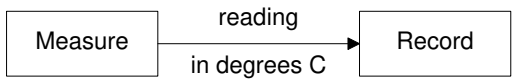
Parameter	Meaning in this example
`above / below`	Keep the two labels on opposite sides of the connector.

Render this diagram

```
tools/render examples/PIC-05-003.pic
```

Try changing: Replace the qualifier with checked value.

Actual output



PIC-05-003 · pic native output

pic · exit 0

PIC-05-004

Emphasize one word

A roff font escape changes emphasis inside the label. Use it sparingly to distinguish an exceptional activity from ordinary steps.

Compare with PIC-05-003 · Label both sides of a connector.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "\f[HB]Review\f[]"
arrow
box "Publish"
.PE
```

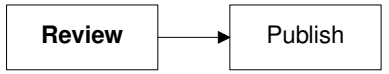
Parameter	Meaning in this example
`\f[HB]`	Select the locally embedded bold font.
`\f[]`	Return to the previous font.

Render this diagram

```
tools/render examples/PIC-05-004.pic
```

Try changing: Make Publish bold instead.

Actual output



PIC-05-004 · pic native output

pic · exit 0

PIC-05-005

Change the label size locally

A size escape changes a label without scaling its box. Use it for a short secondary note while keeping geometric dimensions constant.

Compare with PIC-05-004 · Emphasize one word.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Inspect" "\s[9]visual check\s[0]"
.PE
```

Parameter	Meaning in this example
`\s[9]`	Set the secondary string to nine points.
`\s[0]`	Restore the preceding point size.

Render this diagram

```
tools/render examples/PIC-05-005.pic
```

Try changing: Change the secondary size to 10.

Actual output



PIC-05-005 · pic native output

pic · exit 0

PIC-05-006

Format a computed value

A formatted string turns a number into a concise label. Use it when the displayed value should follow a variable instead of a manually edited caption.

Compare with PIC-05-005 · Change the label size locally.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
n=12
box sprintf("Batch %g",n)
arrow
box "Archive"
.PE
```

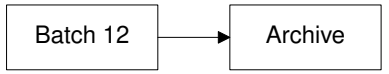
Parameter	Meaning in this example
`sprintf`	Format a string from the supplied expression.

Render this diagram

```
tools/render examples/PIC-05-006.pic
```

Try changing: Change n to 18.

Actual output



PIC-05-006 · pic native output

pic · exit 0

06 · Appearance and emphasis

Use fill, stroke and line style as redundant cues alongside the text.

PIC-06-001

Fill an activity lightly

A shaded fill can group related activities while preserving readable text. Use a light common color for ordinary work.

Compare with PIC-01-001 · Bound an inspection.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Inspect" shaded "wash"
.PE
```

Parameter	Meaning in this example
shaded	Fill the closed shape with a named color.

Render this diagram

```
tools/render examples/PIC-06-001.pic
```

Try changing: Change wash to lightgray.

Actual output



PIC-06-001 · pic native output

pic · exit 0

PIC-06-002

Color an activity outline

An outlined color supplies the pen for the boundary and its label. Use it to distinguish a highlighted activity while leaving the interior unfilled.

Compare with PIC-06-001 · Fill an activity lightly.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Review" outlined "blue"
.PE
```

Parameter	Meaning in this example
`outlined`	Choose the boundary and label pen color.

Render this diagram

```
tools/render examples/PIC-06-002.pic
```

Try changing: Change blue to orange.

Actual output



PIC-06-002 · pic native output

pic · exit 0

PIC-06-003

Thicken an important boundary

Thickness sets the stroke in points. Use a heavier boundary to reinforce a meaning that the label already states.

Compare with PIC-06-002 · Color an activity outline.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Final review" thickness 2
.PE
```

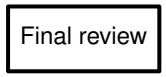
Parameter	Meaning in this example
`thickness 2`	Draw a two-point boundary.

Render this diagram

```
tools/render examples/PIC-06-003.pic
```

Try changing: Reduce the thickness to 1.

Actual output



PIC-06-003 · pic native output

pic · exit 0

PIC-06-004

Dash an optional relation

A dashed stroke can mark an optional or asynchronous relation. State that convention explicitly rather than expecting the line style to carry it alone.

Compare with PIC-06-003 · Thicken an important boundary.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Inspect"
arrow right 1.3 dashed "optional copy" above
box "Audit"
.PE
```

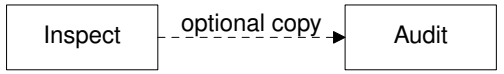
Parameter	Meaning in this example
dashed	Use separated dashes along the connector.

Render this diagram

```
tools/render examples/PIC-06-004.pic
```

Try changing: Change dashed to solid and revise the label.

Actual output



PIC-06-004 · pic native output

pic · exit 0

PIC-06-005

Dot a supporting relation

Dots provide a lighter secondary link. Use them for a reference that is not part of the main process ordering.

Compare with PIC-06-004 · Dash an optional relation.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Record"
line dotted right 1 "reference" above
box "Manual"
.PE
```

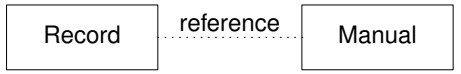
Parameter	Meaning in this example
`dotted`	Use a dotted line without an arrowhead.

Render this diagram

```
tools/render examples/PIC-06-005.pic
```

Try changing: Increase the dotted spacing to .1.

Actual output



PIC-06-005 · pic native output

pic · exit 0

PIC-06-006

Keep scaffolding invisible

An invisible object can reserve a stable position. Use it for layout only, never as a hidden activity in the process model.

Compare with PIC-06-005 · Dot a supporting relation.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect"
Gap: box invis wid .7
B: box "Archive"
arrow from A.e to B.w
.PE
```

Parameter	Meaning in this example
`invis`	Retain object geometry but omit its outline.

Render this diagram

```
tools/render examples/PIC-06-006.pic
```

Try changing: Increase the invisible width to 1.

Actual output



PIC-06-006 · pic native output

pic · exit 0

07 · Variables and expressions

Make dimensions follow named values so a deliberate edit propagates through the diagram.

PIC-07-001

Name a shared width

A dimension variable keeps sibling activities consistent. Use it when several boxes should grow together.

Compare with PIC-02-002 · Stretch a state into an ellipse.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
w=1.2
box wid w "Receive"
arrow
box wid w "Archive"
.PE
```

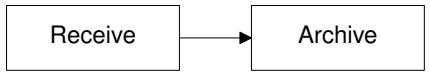
Parameter	Meaning in this example
`w=1.2`	Assign a reusable numeric width.

Render this diagram

```
tools/render examples/PIC-07-001.pic
```

Try changing: Change w to 1.5.

Actual output



PIC-07-001 · pic native output

pic · exit 0

PIC-07-002

Derive the row spacing

Arithmetic can place objects relative to their dimensions. Use an expression to preserve clearance after a width edit.

Compare with PIC-07-001 · Name a shared width.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
w=1.1; gap=.6
A: box wid w "Inspect"
B: box wid w "Store" at A + (w+gap,0)
arrow from A.e to B.w
.PE
```

Parameter	Meaning in this example
w+gap	Combine object width and clear spacing.

Render this diagram

```
tools/render examples/PIC-07-002.pic
```

Try changing: Increase w without changing gap.

Actual output



PIC-07-002 · pic native output

pic · exit 0

PIC-07-003

Read an object dimension

An object exposes its actual width and height. Use those values when later geometry should follow an earlier explicit size.

Compare with PIC-07-002 · Derive the row spacing.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box wid 1.4 ht .6 "Inspect"
B: box wid A.wid ht A.ht "Approve" at A + (2,0)
arrow from A.e to B.w
.PE
```

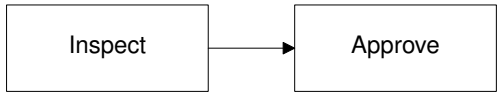
Parameter	Meaning in this example
`A.wid / A.ht`	Read dimensions from an already drawn object.

Render this diagram

```
tools/render examples/PIC-07-003.pic
```

Try changing: Change the first box height to .8.

Actual output



PIC-07-003 · pic native output

pic · exit 0

PIC-07-004

Place a radial note

Sine and cosine convert an angle into an offset. Use them to position a callout around a circular state.

Compare with PIC-07-003 · Read an object dimension.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
r=.9; angle=.65
A: circle "Ready"
P: (r*cos(angle),r*sin(angle))
line from A.ne to P
"checked" ljust at P
.PE
```

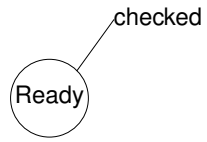
Parameter	Meaning in this example
`sin / cos`	Use radians to calculate the callout position.

Render this diagram

```
tools/render examples/PIC-07-004.pic
```

Try changing: Change angle to 1.

Actual output



PIC-07-004 · pic native output

pic · exit 0

PIC-07-005

Calculate a diagonal distance

A square root converts squared offsets into a length. Use a computed label when a diagram explains its own geometric relation.

Compare with PIC-07-004 · Place a radial note.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
dx=1.6; dy=.8
A: circle rad .12 at (0,0)
B: circle rad .12 at (dx,dy)
line from A to B chop .12
label=sqrt(dx*dx+dy*dy)
sprintf("distance %.2f",label) below at (dx/2,0)
.PE
```

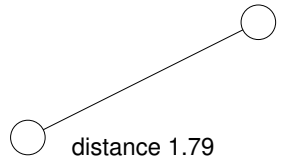
Parameter	Meaning in this example
`sqrt`	Compute the straight-line distance from the offsets.

Render this diagram

```
tools/render examples/PIC-07-005.pic
```

Try changing: Change dy to 1.2.

Actual output



PIC-07-005 · pic native output

pic · exit 0

PIC-07-006

Select a width conditionally

A conditional expression chooses a dimension before drawing. Use it when one configuration flag changes a layout decision.

Compare with PIC-07-005 · Calculate a diagonal distance.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
expanded=1
if expanded then { w=1.8 } else { w=1.1 }
box wid w "Inspection"
.PE
```

Parameter	Meaning in this example
`if / then / else`	Select one branch while constructing the drawing.

Render this diagram

```
tools/render examples/PIC-07-006.pic
```

Try changing: Set expanded to 0.

Actual output



PIC-07-006 · pic native output

pic · exit 0

08 · Repetition and macros

Repeat a geometric idea with explicit parameters instead of duplicating unrelated edits.

PIC-08-001

Repeat a row with a loop

A loop draws a regular row from a numeric range. Use it for a small sequence of batch slots.

Compare with PIC-07-001 · Name a shared width.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
for i=1 to 4 do {
  box wid .65 sprintf("%g",i)
  move right .12
}
.PE
```

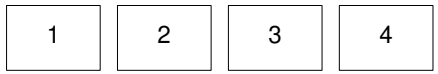
Parameter	Meaning in this example
`for i=1 to 4`	Construct four numbered slots.

Render this diagram

```
tools/render examples/PIC-08-001.pic
```

Try changing: Change the upper limit to 5.

Actual output



PIC-08-001 · pic native output

pic · exit 0

PIC-08-002

Choose the loop step

An explicit step produces a regular subset of values. Use it for evenly spaced markers with meaningful numeric labels.

Compare with PIC-08-001 · Repeat a row with a loop.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
for i=0 to 6 by 2 do {
  circle rad .17 at (i*.35,0)
  sprintf("%g",i) below at (i*.35,-.2)
}
.PE
```

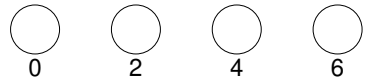
Parameter	Meaning in this example
`by 2`	Advance the loop variable by two each time.

Render this diagram

```
tools/render examples/PIC-08-002.pic
```

Try changing: Change by 2 to by 3.

Actual output



PIC-08-002 · pic native output

pic · exit 0

PIC-08-003

Define an activity macro

A macro names one reusable drawing operation. Use it when several activities should share a boundary and label pattern.

Compare with PIC-08-002 · Choose the loop step.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
define activity { box rad .08 $1 }
activity("Inspect")
arrow
activity("Archive")
.PE
```

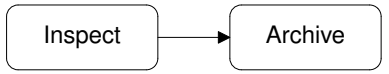
Parameter	Meaning in this example
`define`	Create a reusable source macro.
`$1`	Substitute the first argument.

Render this diagram

```
tools/render examples/PIC-08-003.pic
```

Try changing: Change the radius inside the macro.

Actual output



PIC-08-003 · pic native output

pic · exit 0

PIC-08-004

Pass a geometric parameter

Macro arguments can carry a dimension as well as a label. Use this to vary one feature while keeping the rest of a component stable.

Compare with PIC-08-003 · Define an activity macro.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
define stage { box wid $2 $1 }
stage("Inspect",1.2)
arrow
stage("Archive",1.6)
.PE
```

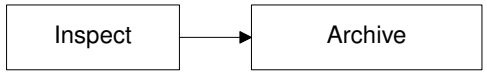
Parameter	Meaning in this example
$2	Use the second macro argument as a width.

Render this diagram

```
tools/render examples/PIC-08-004.pic
```

Try changing: Increase only the second width.

Actual output



PIC-08-004 · pic native output

pic · exit 0

PIC-08-005

Highlight one loop iteration

A conditional inside a loop distinguishes a selected slot. Use a textual or shape convention alongside color when the selection has process meaning.

Compare with PIC-08-004 · Pass a geometric parameter.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
for i=1 to 4 do {
  if i==3 then {
    box wid .65 shaded "wash" "Hold"
  } else {
    box wid .65 sprintf("%g",i)
  }
  move right .12
}
.PE
```

Parameter	Meaning in this example
`i==3`	Choose one repeated slot for the hold label.

Render this diagram

```
tools/render examples/PIC-08-005.pic
```

Try changing: Select slot 2 instead of slot 3.

Actual output



PIC-08-005 · pic native output

pic · exit 0

PIC-08-006

Replace a macro definition

Undefining a macro permits a new local convention. Use this in an explanatory drawing that compares two component styles.

Compare with PIC-08-005 · Highlight one loop iteration.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
define stage { box $1 }
stage("Before")
arrow
undef stage
define stage { box rad .12 $1 }
stage("After")
.PE
```

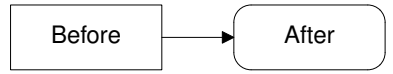
Parameter	Meaning in this example
undef	Remove the previous macro before defining its replacement.

Render this diagram

```
tools/render examples/PIC-08-006.pic
```

Try changing: Change the replacement radius to .2.

Actual output



PIC-08-006 · pic native output

pic · exit 0

09 · Blocks and components

Group related geometry and expose only the attachment points a surrounding diagram needs.

PIC-09-001

Group an intake component

Brackets collect several primitives into one object. Use a block when a small subdiagram must move as a unit.

Compare with PIC-08-003 · Define an activity macro.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Intake: [
  box "Receive"
  arrow
  box "Check"
]
box "Archive" with .w at Intake.e + (.7,0)
arrow from Intake.e to last box.w
.PE
```

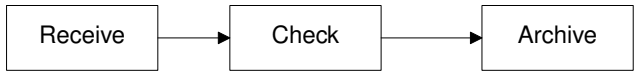
Parameter	Meaning in this example
`[ ... ]`	Create a composite object with its own outer anchors.

Render this diagram

```
tools/render examples/PIC-09-001.pic
```

Try changing: Move the whole Intake block to (0,1).

Actual output



PIC-09-001 · pic native output

pic · exit 0

PIC-09-002

Attach to an internal port

A block can expose a named internal activity. Use a qualified name when a connector must reach a specific part rather than the block boundary.

Compare with PIC-09-001 · Group an intake component.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Unit: [ A: box "Inspect"; arrow; B: box "Store" ]
Audit: box "Audit" at Unit.B + (0,-1)
arrow from Unit.B.s to Audit.n
.PE
```

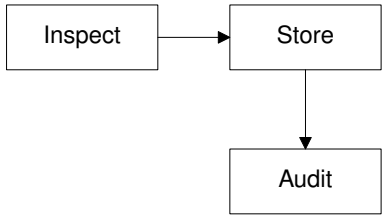
Parameter	Meaning in this example
`Unit.B`	Refer to a named object inside the block.

Render this diagram

```
tools/render examples/PIC-09-002.pic
```

Try changing: Connect the audit to Unit.A instead.

Actual output



PIC-09-002 · pic native output

pic · exit 0

PIC-09-003

Keep a variable local

A block creates a local scope for dimensions. Use it to customize a component without changing the surrounding defaults.

Compare with PIC-09-002 · Attach to an internal port.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
boxwid=1.3
[ boxwid=.7; box "Local" ]
move right .4
box "Outer default"
.PE
```

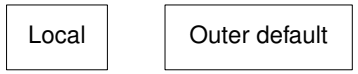
Parameter	Meaning in this example
`Block scope`	Restore the outer boxwid after leaving the brackets.

Render this diagram

```
tools/render examples/PIC-09-003.pic
```

Try changing: Change the local box width to .9.

Actual output



PIC-09-003 · pic native output

pic · exit 0

PIC-09-004

Copy the preceding shape size

Same reuses the previous object of that type. Use it for a pair that should agree in dimensions while keeping distinct labels.

Compare with PIC-09-003 · Keep a variable local.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box wid 1.5 ht .6 "Inspect"
arrow
box same "Approve"
.PE
```

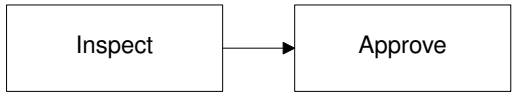
Parameter	Meaning in this example
`same`	Reuse dimensions from the preceding box.

Render this diagram

```
tools/render examples/PIC-09-004.pic
```

Try changing: Change the first height to .8.

Actual output



PIC-09-004 · pic native output

pic · exit 0

PIC-09-005

Nest a reusable subdiagram

A block may contain another block. Use nesting to organize an intake station without flattening its internal structure.

Compare with PIC-09-004 · Copy the preceding shape size.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Station: [
  Desk: [ box "Receive"; arrow; box "Check" ]
  move down .8
  box "Ledger"
]
"Intake station" above at Station.n
.PE
```

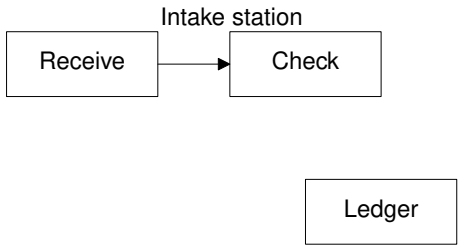
Parameter	Meaning in this example
`Nested brackets`	Keep a smaller component inside a larger one.

Render this diagram

```
tools/render examples/PIC-09-005.pic
```

Try changing: Change Ledger to Register.

Actual output



PIC-09-005 · pic native output

pic · exit 0

PIC-09-006

Frame a composite object

An invisible block boundary can guide a visible frame. Use the block dimensions to leave a consistent margin around its contents.

Compare with PIC-09-005 · Nest a reusable subdiagram.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Unit: [ box "Inspect"; arrow; box "Store" ]
box wid Unit.wid+.35 ht Unit.ht+.45 at Unit
"Archive desk" above at Unit.n + (0,.3)
.PE
```

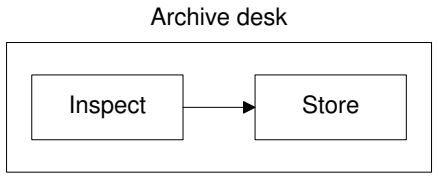
Parameter	Meaning in this example
`Unit.wid / Unit.ht`	Measure the composite object before adding its frame.

Render this diagram

```
tools/render examples/PIC-09-006.pic
```

Try changing: Increase the horizontal margin to .6.

Actual output



PIC-09-006 · pic native output

pic · exit 0

10 · Process patterns

State whether arrows mean ordering, alternatives or required parallel work; pic draws the supplied convention.

PIC-10-001

Branch on a check

Two labeled outgoing arrows express exclusive alternatives in this example. Use them when exactly one outcome follows the check.

Compare with PIC-04-001 · Route through an elbow.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box wid 1.5 "Valid record?"
B: box "Archive" at (3,.7)
C: box "Return" at (3,-.7)
arrow from A.ne to B.w "yes" above
arrow from A.se to C.w "no" below
.PE
```

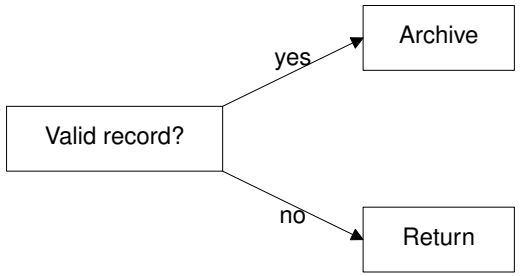
Parameter	Meaning in this example
`Labeled branches`	Make the exclusive alternatives explicit.

Render this diagram

```
tools/render examples/PIC-10-001.pic
```

Try changing: Change the question and both outcome labels.

Actual output



PIC-10-001 · pic native output

pic · exit 0

PIC-10-002

Merge alternative paths

Two incoming paths can meet at one activity after an exclusive choice. Use a merge only when it does not imply waiting for both alternatives.

Compare with PIC-10-001 · Branch on a check.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "New record" at (0,.7)
B: box "Correction" at (0,-.7)
C: box "Inspect" at (2,0)
arrow from A.e to C.nw
arrow from B.e to C.sw
"either input" above at C.n
.PE
```

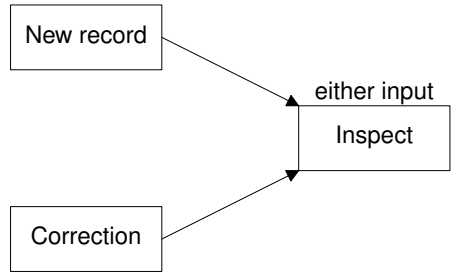
Parameter	Meaning in this example
`either input`	State that the merge does not wait for both paths.

Render this diagram

```
tools/render examples/PIC-10-002.pic
```

Try changing: Replace Correction with Reopened record.

Actual output



PIC-10-002 · pic native output

pic · exit 0

PIC-10-003

Require both parallel checks

A fork and join can express required parallel work when the rule is stated. Here release waits for both the label and condition checks.

Compare with PIC-10-002 · Merge alternative paths.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Fork: line from (0,.7) to (0,-.7) thickness 3
A: box wid 1.6 "Check label" at (1.2,.55)
B: box wid 1.6 "Check condition" at (1.2,-.55)
Join: line from (2.4,.7) to (2.4,-.7) thickness 3
arrow from (0,.55) to A.w
arrow from (0,-.55) to B.w
arrow from A.e to (2.4,.55)
arrow from B.e to (2.4,-.55)
"both required" above at (1.2,1)
.PE
```

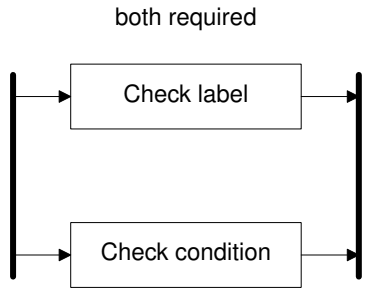
Parameter	Meaning in this example
`Thick bars`	Use the stated AND-fork and AND-join convention.

Render this diagram

```
tools/render examples/PIC-10-003.pic
```

Try changing: Change the two required checks.

Actual output



PIC-10-003 · pic native output

pic · exit 0

PIC-10-004

Bound a retry route

A retry must state its limit instead of implying endless repetition. Use a separate escalation outcome after the allowed correction attempt.

Compare with PIC-10-003 · Require both parallel checks.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect"
B: box "Correct" at (1.8,-1)
C: box "Escalate" at (3.6,0)
arrow from A.se to B.nw
"fail" ljust at (1.4,-.45)
arrow from B.w left .7 then to A.s
arrow from A.e to C.w "second failure" above
"retry once" below at B.s + (0,-.12)
.PE
```

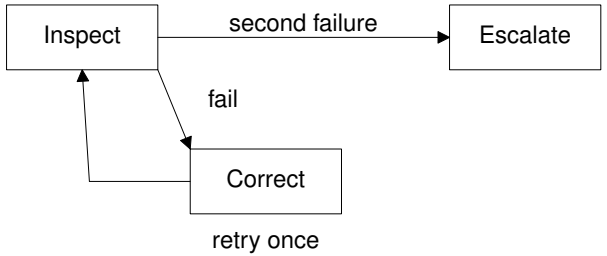
Parameter	Meaning in this example
`retry once`	Supply a finite process rule; pic does not enforce it.

Render this diagram

```
tools/render examples/PIC-10-004.pic
```

Try changing: Change once to twice and revise the escalation label.

Actual output



PIC-10-004 · pic native output

pic · exit 0

PIC-10-005

Show an ownership handoff

A divider distinguishes responsibility from sequence. Use explicit role labels so the crossing identifies a handoff between people.

Compare with PIC-10-004 · Bound a retry route.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Inspect" at (0,0)
B: box "Archive" at (2.3,0)
line from (1.15,.8) to (1.15,.3) dashed
line from (1.15,-.3) to (1.15,-.8) dashed
"Technician" at (0,1)
"Curator" at (2.3,1)
arrow from A.e to B.w "record" above
.PE
```

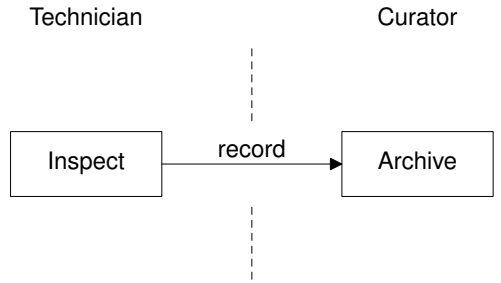
Parameter	Meaning in this example
`Dashed divider`	Separate owners without treating the line as a process step.

Render this diagram

```
tools/render examples/PIC-10-005.pic
```

Try changing: Rename the two roles.

Actual output



PIC-10-005 · pic native output

pic · exit 0

PIC-10-006

Separate failure from completion

A failure branch should not disappear into the successful outcome. Use a distinct held state when the archive must retain an unresolved record.

Compare with PIC-10-005 · Show an ownership handoff.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Review"
B: ellipse wid 1.4 ht .55 "Released" at (3,.65)
C: ellipse wid 1.4 ht .55 "Held" at (3,-.65)
arrow from A.ne to B.w
"pass" at (1.6,.7)
arrow from A.se to C.w
"fail" at (1.6,-.7)
.PE
```

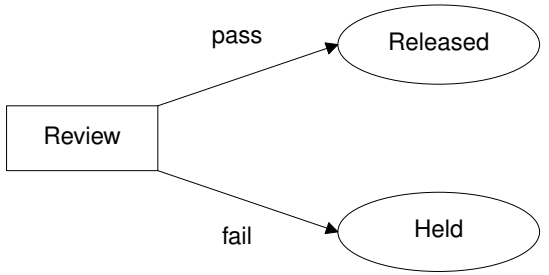
Parameter	Meaning in this example
`Distinct states`	Keep failure and successful completion visible.

Render this diagram

```
tools/render examples/PIC-10-006.pic
```

Try changing: Change Held to Needs correction.

Actual output



PIC-10-006 · pic native output

pic · exit 0

11 · Technical relationships

Reuse the same geometry while naming whether a connector means a call, dependency, transition or pointer.

PIC-11-001

Map a component call

A directed connector may represent a call rather than elapsed process order. Use a verb on the edge when the boxes name software components.

Compare with PIC-01-003 · Name the handoff.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Intake app"
arrow right 1 "writes" above
box wid 1.4 "Record store"
.PE
```

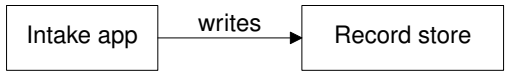
Parameter	Meaning in this example
`writes`	State the component relationship explicitly.

Render this diagram

```
tools/render examples/PIC-11-001.pic
```

Try changing: Change writes to reads and reverse the arrow.

Actual output



PIC-11-001 · pic native output

pic · exit 0

PIC-11-002

Draw a state transition

States and transition labels answer different questions from activity boxes. Use event names on arrows between circular states.

Compare with PIC-11-001 · Map a component call.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: circle "New"
B: circle "Filed" at (1.8,0)
arrow from A.e to B.w "accept" above
.PE
```

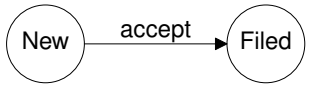
Parameter	Meaning in this example
accept	Name the event that changes the state.

Render this diagram

```
tools/render examples/PIC-11-002.pic
```

Try changing: Change accept to approve.

Actual output



PIC-11-002 · pic native output

pic · exit 0

PIC-11-003

Trace a file lifecycle

A document can move through several named versions. Use consistent boundaries and explicit revision labels to distinguish identity from action.

Compare with PIC-11-002 · Draw a state transition.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Draft.csv"
arrow right .9 "validate" above
box "Checked.csv"
arrow right .9 "freeze" above
box "Release.csv"
.PE
```

Parameter	Meaning in this example
`Version labels`	Track file states across the lifecycle.

Render this diagram

```
tools/render examples/PIC-11-003.pic
```

Try changing: Change freeze to sign off.

Actual output



PIC-11-003 · pic native output

pic · exit 0

PIC-11-004

Map a dependency fan-in

Several inputs may be prerequisites of one artifact. Use arrows toward the artifact and a caption stating that all inputs are required.

Compare with PIC-11-003 · Trace a file lifecycle.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Readings" at (0,.7)
B: box "Metadata" at (0,-.7)
C: box "Report" at (2,0)
arrow from A.e to C.nw
arrow from B.e to C.sw
"both inputs required" below at C.s + (0,-.5)
.PE
```

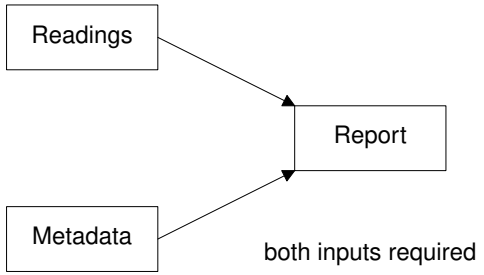
Parameter	Meaning in this example
`Fan-in`	Show supplied prerequisites rather than a scheduled execution.

Render this diagram

```
tools/render examples/PIC-11-004.pic
```

Try changing: Add a third Style input above the report.

Actual output



PIC-11-004 · pic native output

pic · exit 0

PIC-11-005

Lay out queue slots

Repeated cells show the position of items in a queue. Use labels for the head and tail so left-to-right placement has an explicit interpretation.

Compare with PIC-11-004 · Map a dependency fan-in.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
for i=1 to 4 do { box wid .65 sprintf("R%g",i) }
"head" below at 1st box.s + (0,-.12)
"tail" below at last box.s + (0,-.12)
.PE
```

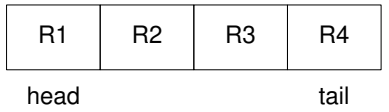
Parameter	Meaning in this example
`1st box / last box`	Refer to the first and latest cells without naming each one.

Render this diagram

```
tools/render examples/PIC-11-005.pic
```

Try changing: Increase the queue to five slots.

Actual output



PIC-11-005 · pic native output

pic · exit 0

PIC-11-006

Arrange a small hierarchy

A parent can connect to children placed on one row. Use this for containment or responsibility, and name that relation in the surrounding explanation.

Compare with PIC-11-005 · Lay out queue slots.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Archive" at (0,1)
B: box "Field A" at (-1,0)
C: box "Field B" at (1,0)
line from A.s to B.n
line from A.s to C.n
.PE
```

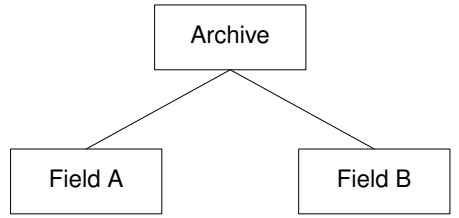
Parameter	Meaning in this example
`Shared parent`	Represent the stated containment hierarchy.

Render this diagram

```
tools/render examples/PIC-11-006.pic
```

Try changing: Rename the child collections.

Actual output



PIC-11-006 · pic native output

pic · exit 0

12 · Finished diagrams

Combine established primitives into small original diagrams with an explicit communication goal.

PIC-12-001

Finish an intake overview

A compact overview keeps the main path readable. Use a separate note for the shared identifier instead of repeating a long label in every box.

Compare with PIC-11-003 · Trace a file lifecycle.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box rad .08 "Receive"
arrow
box rad .08 "Inspect"
arrow
box rad .08 "Archive"
"One record ID follows the sample" below at (1.6,-.7)
.PE
```

Parameter	Meaning in this example
`Rounded activities`	Apply one consistent boundary style to the main path.

Render this diagram

```
tools/render examples/PIC-12-001.pic
```

Try changing: Replace Inspect with two short named checks.

Actual output



PIC-12-001 · pic native output

pic · exit 0

PIC-12-002

Finish a review feedback diagram

A review loop benefits from a clear forward route and a named return. Use different vertical bands so the correction path does not obscure publication.

Compare with PIC-12-001 · Finish an intake overview.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box "Draft"
arrow
B: box "Review"
arrow right .9 "accept" above
box "Release"
arrow from B.s down .6 then left B.x-A.x then to A.s
"revise" at ((A.x+B.x)/2,-1)
.PE
```

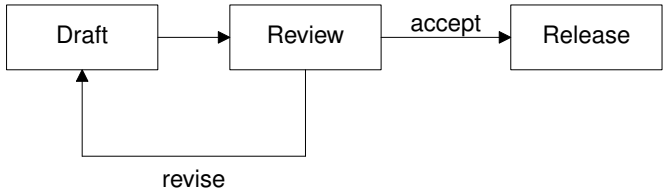
Parameter	Meaning in this example
`Separate return band`	Keep the feedback path clear of the successful route.

Render this diagram

```
tools/render examples/PIC-12-002.pic
```

Try changing: Change revise to correct metadata.

Actual output



PIC-12-002 · pic native output

pic · exit 0

PIC-12-003

Finish a record handoff

The record passed between owners deserves a label as well as a direction. Use a two-line activity label to keep the role and action distinct.

Compare with PIC-12-002 · Finish a review feedback diagram.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
box "Technician" "Inspect"
arrow right 1.7 "checked record" above
box "Curator" "Archive"
.PE
```

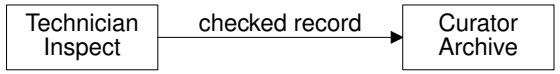
Parameter	Meaning in this example
`Two-line role/action label`	State who performs the work and what they do.

Render this diagram

```
tools/render examples/PIC-12-003.pic
```

Try changing: Change the delivered item to signed record.

Actual output



PIC-12-003 · pic native output

pic · exit 0

PIC-12-004

Finish a publication pipeline

A named macro keeps repeated publication stages consistent. Use it to make a later style change affect the whole path.

Compare with PIC-12-003 · Finish a record handoff.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
define stage { box wid 1.1 rad .08 shaded "wash" $1 }
stage("Validate")
arrow
stage("Summarize")
arrow
stage("Publish")
.PE
```

Parameter	Meaning in this example
`Shared stage macro`	Apply the same presentation to every publication activity.

Render this diagram

```
tools/render examples/PIC-12-004.pic
```

Try changing: Change the macro fill to lightgray.

Actual output



PIC-12-004 · pic native output

pic · exit 0

PIC-12-005

Add a dimension annotation

A dimension line explains geometry without becoming a process edge. Use extension lines and an explicit unit so the measurement cannot be read as a handoff.

Compare with PIC-12-004 · Finish a publication pipeline.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
A: box wid 2 ht .6 "Inspection desk"
line from A.sw down .5
line from A.se down .5
line <-> from A.sw + (0,-.4) to A.se + (0,-.4)
sprintf("%g inches",A.wid) below at A.s + (0,-.58)
.PE
```

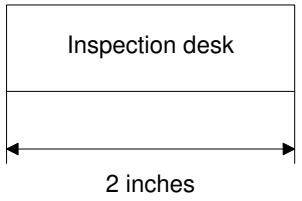
Parameter	Meaning in this example
`Dimension line`	Annotate physical width separately from process meaning.

Render this diagram

```
tools/render examples/PIC-12-005.pic
```

Try changing: Change the box width to 2.5; confirm that the numeric label follows.

Actual output



PIC-12-005 · pic native output

pic · exit 0

PIC-12-006

Compare direction within one composition

Two block instances can present the same stages in different orientations. Use a named block to keep each alternative independent and clearly labeled.

Compare with PIC-12-005 · Add a dimension annotation.

Complete pic source

```
.so config/render.roff
.PS
copy "config/style.pic"
Horizontal: [ right; box "Inspect"; arrow; box "Store" ]
Vertical: [ down; box "Inspect"; arrow; box "Store" ] with .nw at Horizontal.sw + (0,-.7)
"Across" above at Horizontal.n
"Down" ljust at Vertical.e + (.2,0)
.PE
```

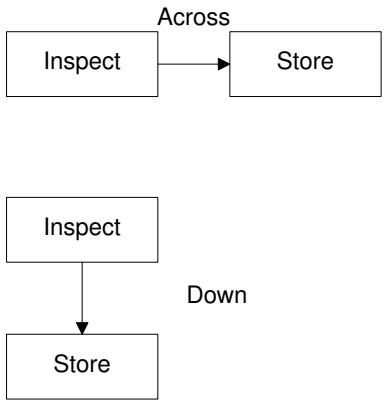
Parameter	Meaning in this example
`Block-local direction`	Keep the two reading directions independent.

Render this diagram

```
tools/render examples/PIC-12-006.pic
```

Try changing: Change only the vertical block to Receive and Check.

Actual output



PIC-12-006 · pic native output

pic · exit 0

Choosing a diagram

Start with the question and the relationship to communicate. Use the smallest example that contains the needed idea, then add labels, grouping and styling only where they clarify it. Inspect the native figure at its intended reading size. The examples draw supplied process meanings and statistics; a drawing does not execute a workflow or establish a causal claim.

Goal	Start here
Arrange one short process	Chapter 01
Place objects with named anchors	Chapter 03
Route a return connection	Chapter 04
Reuse a component	Chapter 08
Explain choices and required parallel work	Chapter 10
Finish a small technical diagram	Chapter 12

Feature coverage

Feature family	Chapter
First diagrams	01
Shapes and visual roles	02
Anchors and alignment	03
Connections and routes	04
Labels and typography	05
Appearance and emphasis	06
Variables and expressions	07
Repetition and macros	08
Blocks and components	09
Process patterns	10
Technical relationships	11
Finished diagrams	12

References and licensing

GNU groff pictures and the installed `doc/groff-1.25.0/pic.ms` / `pic.pdf` manual.

For named process shapes and automatic layout, compare the D2 tour, Graphviz tour and ditaa tour. The groff tour places pic drawings inside documents.

Original prose, programs and synthetic datasets are CC0. Copyrightable structure and the publishing framework are CC BY 4.0, Lukas C. Bossert. Bundled fonts, package sources and other third-party material retain their own terms; consult the license scope and `LICENSES/`. The shared example and command index links every rendered program below.

Example and command index

- PIC-01-001 · Bound an inspection
- PIC-01-002 · Connect the next action
- PIC-01-003 · Name the handoff
- PIC-01-004 · Stack a process vertically
- PIC-01-005 · Split a long label
- PIC-01-006 · Explain the source
- PIC-02-001 · Mark a state with a circle
- PIC-02-002 · Stretch a state into an ellipse
- PIC-02-003 · Round an activity boundary
- PIC-02-004 · Construct a decision diamond
- PIC-02-005 · Separate a boundary from a connection
- PIC-02-006 · Bend a relation smoothly
- PIC-03-001 · Name the objects
- PIC-03-002 · Align an edge to an anchor
- PIC-03-003 · Offset a related activity
- PIC-03-004 · Share a horizontal coordinate
- PIC-03-005 · Place a note between activities
- PIC-03-006 · Save the current position
- PIC-04-001 · Route through an elbow
- PIC-04-002 · Show a two-way exchange
- PIC-04-003 · Reverse the arrowhead
- PIC-04-004 · Stop at circular boundaries
- PIC-04-005 · Arc around a state
- PIC-04-006 · Route a rework loop below
- PIC-05-001 · Place a label above a line
- PIC-05-002 · Align a note to the left
- PIC-05-003 · Label both sides of a connector
- PIC-05-004 · Emphasize one word
- PIC-05-005 · Change the label size locally
- PIC-05-006 · Format a computed value
- PIC-06-001 · Fill an activity lightly
- PIC-06-002 · Color an activity outline
- PIC-06-003 · Thicken an important boundary
- PIC-06-004 · Dash an optional relation
- PIC-06-005 · Dot a supporting relation
- PIC-06-006 · Keep scaffolding invisible
- PIC-07-001 · Name a shared width
- PIC-07-002 · Derive the row spacing
- PIC-07-003 · Read an object dimension
- PIC-07-004 · Place a radial note
- PIC-07-005 · Calculate a diagonal distance
- PIC-07-006 · Select a width conditionally
- PIC-08-001 · Repeat a row with a loop
- PIC-08-002 · Choose the loop step
- PIC-08-003 · Define an activity macro
- PIC-08-004 · Pass a geometric parameter
- PIC-08-005 · Highlight one loop iteration
- PIC-08-006 · Replace a macro definition
- PIC-09-001 · Group an intake component
- PIC-09-002 · Attach to an internal port
- PIC-09-003 · Keep a variable local
- PIC-09-004 · Copy the preceding shape size
- PIC-09-005 · Nest a reusable subdiagram
- PIC-09-006 · Frame a composite object

- PIC-10-001 · Branch on a check
- PIC-10-002 · Merge alternative paths
- PIC-10-003 · Require both parallel checks
- PIC-10-004 · Bound a retry route
- PIC-10-005 · Show an ownership handoff
- PIC-10-006 · Separate failure from completion
- PIC-11-001 · Map a component call
- PIC-11-002 · Draw a state transition
- PIC-11-003 · Trace a file lifecycle
- PIC-11-004 · Map a dependency fan-in
- PIC-11-005 · Lay out queue slots
- PIC-11-006 · Arrange a small hierarchy
- PIC-12-001 · Finish an intake overview
- PIC-12-002 · Finish a review feedback diagram
- PIC-12-003 · Finish a record handoff
- PIC-12-004 · Finish a publication pipeline
- PIC-12-005 · Add a dimension annotation
- PIC-12-006 · Compare direction within one composition

Content: CC0 1.0. Structure and publishing framework: CC BY 4.0 by Lukas C. Bossert. Repository source · Publishing framework source · License scope and third-party notices.