

GUI Designer Guide

From first form to connected applications



LEARN	BUILD	REFERENCE
Small examples	Real Kokum interfaces	Searchable A-Z entries

Linux 1.1.0 • VM 1.7 • GUI contract 1.9
September 2026

The language designed and developed by Anil Jagtap.
Email: anil.softx@gmail.com

Includes DatePicker.GetDate, TRIM / LTRIM / RTRIM,
REST requests, TOTP and the current Linux distribution workflow.

Contents

Use the linked entries below or the document navigation pane. The A-Z index at the end links directly to individual entries.

About this edition	3
Install and start the Linux tools	4
Network IDE and user administration	5
Create your first Designer application	7
Work efficiently in the Designer	9
Signals, slots and generated state	11
Layouts, themes and multiple forms	13
Practical labs	15
Widget reference A-Z	19
Common properties and event directory	59
Troubleshooting and known boundaries	60
Sources, scope and verification	61
A-Z widget and topic index	63

About this edition

This is the current **Kokum Linux 1.1 developer edition**, including string trimming, the Designer rename correction, DatePicker and DatePicker.GetDate. It is written for users of the supplied Linux distribution; you do not need to compile Kokum itself.

Two books, complementary purposes

The GUI Designer Guide explains building interfaces and every entry in the current widget schema. The Function Reference is an A-Z catalogue of all 110 public global built-ins, 112 resource methods and 22 control-proxy methods, with a separate property directory. Neither book treats internal C entry points as user-facing functions.

Value / artifact	Meaning in this edition
1.1.0	Product label. Do not use this label alone as a runtime-compatibility check.
VM ABI 1.7	Needed by the new trimming built-ins; older-only applications retain their lower requirements.
GUI contract 1.9	Newly compiled DatePicker forms, including GetDate support.
.kokum / .kform	Language source / form source.
.kbc / .kres / .kapp	Compiled module / compiled form / complete application bundle.

Example conventions

A **Main body** listing goes inside FUNCTION Main AS INTEGER and ends with RETURN 0 / ENDFUNC. A **slot body** goes inside a bound procedure with sender, event and form parameters. SET-* labels identify reusable declarations, controls or external configuration. The reference supplies each setup explicitly; a context-dependent fragment is not presented as an independent console program.

Square brackets in a syntax signature mean optional arguments; they are not literal characters to copy. In executable snippets, brackets create or index real Kokum arrays. Kokum array indices are one-based; many GUI selection and TableView method indices are zero-based. Empty STRING and NULL are different values.

Known limitations are kept visible

The current shared GUI findings are TableView automatic-array-to-method handoff, programmatic Splitter position, one TextBox Visible=false path, and dynamic Image refresh. ModalDialog and SubFormHost are documented as restricted/legacy schema entries, not newly certified widgets. See the troubleshooting chapter.

Source references such as [S02] identify the authoritative files listed at the end. Screenshots from the earlier Designer manual are labelled as retained interface examples; they illustrate layout and workflow, not a new all-widget acceptance test.

Install and start the Linux tools

Obtain the publisher-supplied kokum.tar.gz or the complete dist directory. Keep bin/, lib/ and share/ together. Do not copy an individual .elf file and expect it to locate the bundled libraries and Designer assets.

Precompiled distribution

```
tar -xzf kokum.tar.gz
cd kokum
./bin/kokumc --version
./bin/kokumvm --version
./start_ide.sh
```

The launch scripts choose the local libraries and assets. A compatible Linux CPU/glibc baseline, a browser, ordinary shell facilities and the operating system's certificate/random-device facilities are still required. This is not a Windows/macOS installation guide or a universal Linux binary guarantee. [S15]

Open an existing project

Linux terminal

```
./bin/kokum-ide /absolute/path/to/your/project
```

Keep projects in a writable user directory, outside the installed runtime. Start the IDE with the project path, or open a project from its workspace controls. Normal local mode is single-user. Do not expose its local host directly as a public server.

Create and run a small console application

Save as hello.kokum in a working directory

```
FUNCTION Main AS INTEGER
  LOCAL name AS STRING
  name = TRIM(" Kokum ")
  ? "Hello, " + name
  RETURN 0
ENDFUNC
```

Use full bin/ paths unless they are already on PATH

```
kokumc check hello.kokum
kokumc run-bytecode hello.kokum
kokumc compile hello.kokum -o hello.kbc
kokumvm hello.kbc
```

Expected output is Hello, Kokum. These commands compile your application, not Kokum's compiler or IDE. Publisher source-building and distribution generation are intentionally outside the end-user workflow.

Network IDE and user administration

The network IDE is one Linux-hosted installation with authenticated, per-user workspaces. It uses the existing LAN server configuration and users.csv account file. The documented mode uses ordinary HTTP and plaintext account storage, so restrict it to a trusted private LAN and keep configuration private. It is not an Internet-ready identity service. [S14]

Prepare a private state directory

Linux terminal

```
mkdir -p "$HOME/kokum-lan"
chmod 700 "$HOME/kokum-lan"
```

Create users.csv inside that directory with the header below. Replace the example value locally with a strong, unique password before starting the server. Never publish this file or place it inside an application bundle.

users.csv — replace the password before use

```
username,password
admin,REPLACE_WITH_YOUR_PRIVATE_PASSWORD
```

Linux terminal

```
chmod 600 "$HOME/kokum-lan/users.csv"
```

Configure the host

Use the actual private IPv4 address assigned to the Linux host and the matching client subnet. The following is a template, not a claim that your server uses this address. Substitute an absolute base_directory path; do not leave /home/yourname unchanged.

private-ide-lan.conf

```
[server]
format_version=2
mode=lan
listen_address=192.168.1.20
listen_port=9440
advertised_host=192.168.1.20
allowed_client_cidr=192.168.1.0/24
allow_loopback=true
base_directory=/home/yourname/kokum-lan
users_file=users.csv
workspace_directory=workspaces
tvm_path=auto
enable_project_check=true
enable_project_build=true
enable_project_run=true
enable_project_deploy=true
```

Linux terminal

```
./start_network_ide.sh /absolute/path/private-ide-lan.conf
```

Open <http://192.168.1.20:9440> in a client browser after substituting the real host address. The listener requires an explicit private interface; do not substitute 0.0.0.0 or open a public firewall port. Restrict the allowed client range to the intended network.

Create and manage users

Sign in as admin and open the administrator user controls. Create a separate account for each developer, choose a private password and distribute it through an appropriate private channel. The reserved admin account manages users; other accounts are developers. Account names are checked case-insensitively. Keep users.csv writable only by the server owner.

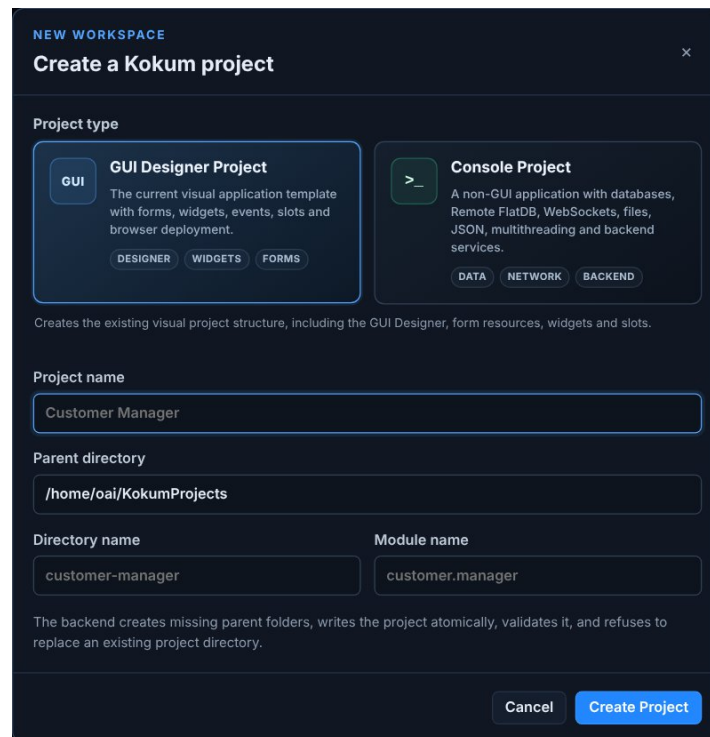
The administrator can list/create users, change passwords and delete permitted non-admin accounts. Credential/account changes invalidate affected sessions. Do not delete or share the admin identity. Use the server's configured session, lockout and workspace limits instead of relying on browser-side restrictions.

Location / operation	Behavior
workspaces/<username>/projects	User-owned projects.
state, cache, temp and trash	Per-user server workspace support directories.
Check / Build	Current authenticated project operations; controlled by server configuration.
Run / Deploy	Use the configured server-side VM and Linux environment.
Preview / transfer	Server-side Run is not automatic browser-preview routing to a client, and a produced artifact is not automatically downloaded.

Stop the host before manually editing account files. Keep private backups and verify permissions after restoring them. No ready-made admin account, live users file or private configuration is shipped in the runtime distribution. [S14, S15]

Create your first Designer application

Choose New Project → GUI Designer Project. Supply an application name and a new writable project directory. The wizard creates the manifest, main module, slots module, form and browser resources. Open `ui/main_window.kform` in the project Explorer.



Retained New Project wizard from the supplied GUI Designer manual. Field arrangement may vary with the frontend.

File	Responsibility
<code>kokum.toml</code>	Modules, entry routine, forms and packaged resources.
<code>ui/main_window.kform</code>	Control IDs, properties, hierarchy and event targets.
<code>src/slots.kokum</code>	Exports, generated declarations and programmer-owned procedures.
<code>web/</code>	HTML/CSS/JavaScript and application assets included when bundling.
<code>build/</code> and <code>dist/</code>	Generated application output; not authoritative editable source.

A command-line application wizard is also available:

Linux terminal

```
kokumc project new --type web-gui \
  --name HelloDesigner --directory /absolute/path/HelloDesigner
```

Design Hello Designer

Place `txtName` (TextBox), `chkUpper` (CheckBox), `lblResult` (Label), `btnGreet` and `btnClear` (Buttons). Give them useful captions, then connect `btnGreet.clicked` and `txtName.enterPressed` to the same exported procedure. Connect `btnClear.clicked` to a separate reset procedure.

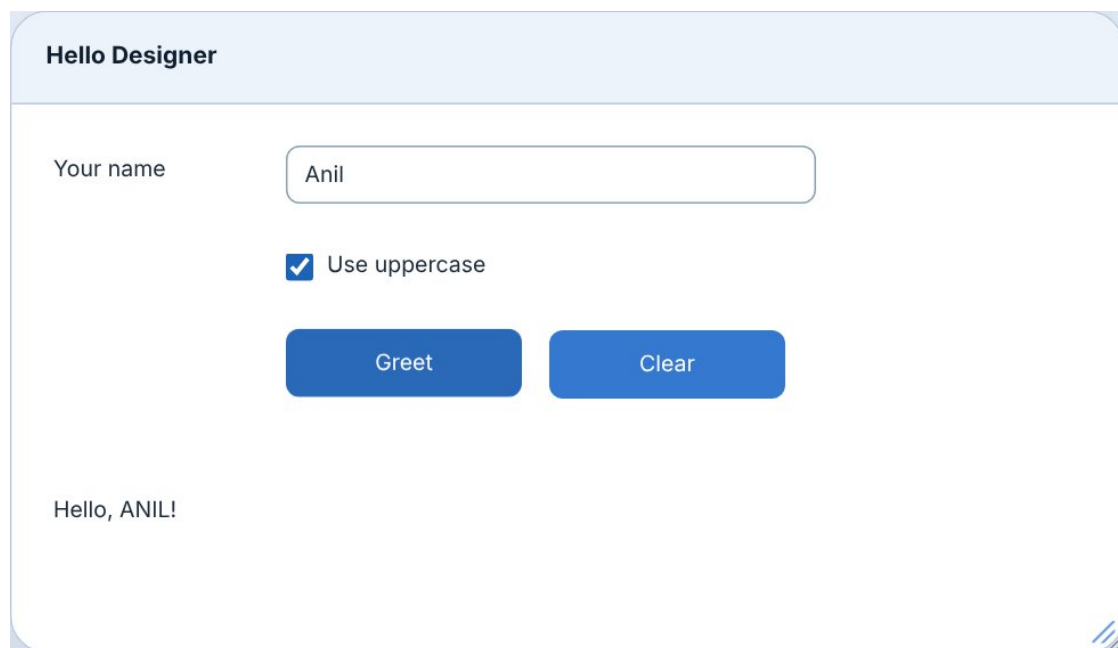
Add these routines beneath the Designer-owned generated block

```
EXPORT PROCEDURE btnGreet_clicked
EXPORT PROCEDURE btnClear_clicked

PROCEDURE btnGreet_clicked(sender, event, form)
  LOCAL greeting AS STRING
  greeting = "Hello, " + TRIM(txtName)
  IF chkUpper
    greeting = UPPER(greeting)
  ENDIF
  form.lblResult.Text = greeting
ENDPROC

PROCEDURE btnClear_clicked(sender, event, form)
  txtName = ""
  chkUpper = .F.
  form.lblResult.Text = "Ready"
ENDPROC
```

The Designer generates PUBLIC txtName AS STRING and PUBLIC chkUpper AS LOGICAL. Do not paste duplicate declarations outside its marked block. Set Text to Ready on lblResult in the Properties panel. Run, type a name and click Greet; checking chkUpper changes the greeting to uppercase.



Retained Hello Designer runtime example. It illustrates the same input → slot → output pattern.

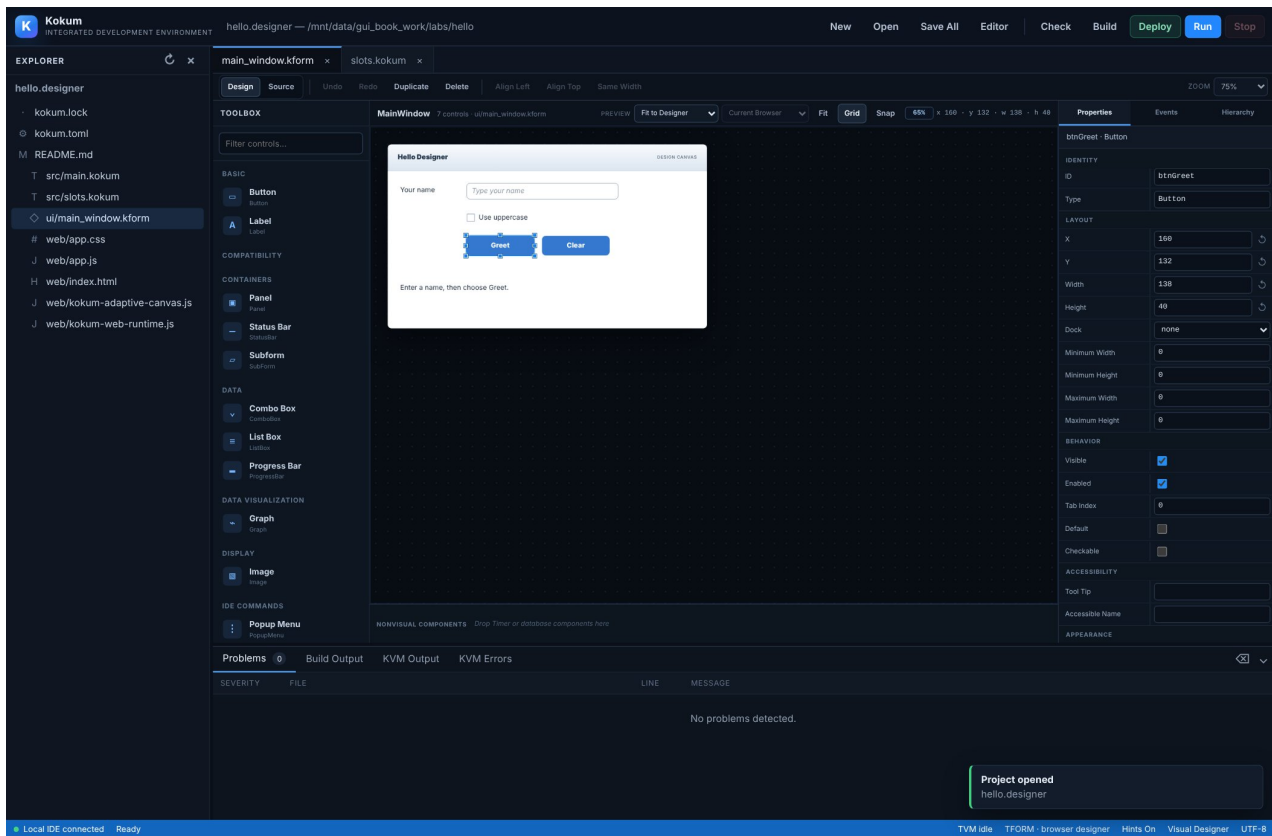
Check, bundle and run

Linux terminal

```
kokumc project check /absolute/path/HelloDesigner/kokum.toml
kokumc bundle /absolute/path/HelloDesigner/kokum.toml
kokumvm /absolute/path/HelloDesigner/dist/your-application.kapp
```

Use the actual bundle filename reported by the project manifest/build. The compiler checks declarations and bindings; a successful visual preview alone does not prove that a slot is callable. Save the project before deploying. [S03]

Work efficiently in the Designer



Retained Designer workspace: Explorer, canvas, Toolbox and inspector. The current DatePicker is an additional Input control.

The four views to keep in agreement

Explorer selects the file. The canvas shows geometry. Hierarchy establishes actual parent/child structure. Properties and Events describe the selected node. A control that appears inside a Panel can still be a sibling unless its parent is correct; check Hierarchy rather than relying on appearance.

Action	Workflow
Add a control	Drag it from Toolbox, then set a stable ID and meaningful properties.
Select / arrange	Use canvas selection or Hierarchy; use align/distribute controls where offered.
Move / resize	Arrow keys move the selection; Shift with arrows adjusts size.
Copy / duplicate	Ctrl+C / Ctrl+V use the Designer clipboard; Ctrl+D duplicates. New IDs must remain unique.
Pan / zoom	Space-drag pans; Ctrl+wheel zooms the canvas.
Grid and snap	Grid visibility and snapping are separate settings. Shift temporarily enables snap; Alt bypasses it while manipulating controls.
Undo / redo	Use the Designer history and verify dependent generated code after an accepted change.

Properties and collection editors

Use the property editor suited to the value: checkboxes for LOGICAL settings, enum lists for fixed choices, date fields for DatePicker value/bounds, and collection dialogs for Items/Columns/Rows or hierarchical data. Tree indentation is meaningful. TabPage Text is the tab caption; it is not a declaration of a new scope.

Keep IDs, visible Text/captions, project/module names and slot names distinct. Changing a caption does not rename a variable; changing an ID does not safely rewrite arbitrary application strings or custom procedure names.

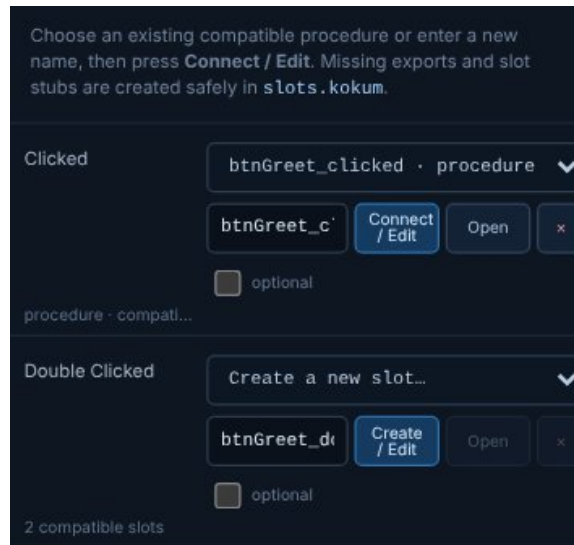
Safe control renaming

Rename comboBox1 to cmbFeeTerm in the Designer, then save. The form and generated variable block must both identify cmbFeeTerm. A true case-insensitive collision across forms sharing a slots module is rejected. The current fix prevents rejected rename state from later being saved and excludes the control itself from duplicate detection.

Update programmer-written references to the old ID deliberately. A blocked rename must not be “fixed” by keeping competing PUBLIC declarations. Preserve comments, imports, exports and unrelated arrays. Back up a mismatched project, review IDs/references, then use project check to regenerate the managed block. [S03]

Signals, slots and generated state

A signal is an event emitted by a control. A slot is an exported Kokum routine selected as that event's handler. The binding lives in the .kform; the implementation lives in the configured slots module. Creating a procedure alone does not establish an event binding.



Retained Events inspector. Select the control and signal, then create/connect or open the slot.

Kokum example

```
EXPORT PROCEDURE btnSave_clicked
PROCEDURE btnSave_clicked(sender, event, form)
    form.lblStatus.Text = "Saved for: " + TRIM(txtName)
ENDPROC
```

Parameter	Role
sender	Identity/state of the control that raised the event.
event	Event-specific values, such as Value/Text, key data or a close-cancellation flag.
form	Control-proxy MAP for this form. Use form.lblStatus.Text or form.dtAdmission.GetDate(...).

The binding checker expects the exported callable in the named module and the standard sender, event, form signature. Connect/Edit creates or opens the routine. Disconnect removes the binding; it is not permission to delete unrelated user code. [S03]

Automatic variables versus control objects

Control	Generated value	Object / resource access
TextBox, TextArea, ComboBox, DatePicker	STRING	form.<id> for properties and supported control methods
CheckBox, RadioButton	LOGICAL	form.<id> for control properties
ListBox, ListView, TreeView, TableView	ARRAY	form.<id> for the applicable native property/method surface
Graph	GRAPH	<id>.SetData(...), not a generic STRING
Image	IMAGE	<id>.SetImage(...), native resource
SQLite / TigerDB / PostgreSQL	DATABASE	<id>.Query(...), configured connection

Control	Generated value	Object / resource access
WebSocketClient	WEBSOCKET	<id>.Connect()/Receive(), owner resource
Label, Button, Timer, containers	No automatic scalar value	Use form.<id> or form actions

Kokum example

```
&& <KOKUM-AUTO-WIDGET-VARIABLES-BEGIN>
&& Generated from .kform input controls. Do not edit this block.
PUBLIC chkUpper AS LOGICAL
PUBLIC dtAdmission AS STRING
PUBLIC txtName AS STRING
&& <KOKUM-AUTO-WIDGET-VARIABLES-END>
```

The toolchain owns only its marked generated regions. Custom declarations such as PUBLIC names AS ARRAY remain outside those regions. A declaration does not instantiate a widget: the Designer model and hosting session supply its control/resource. IDs are case-insensitively unique across forms sharing a slots module.

When values synchronize

Browser input is synchronized before a bound slot runs. The slot reads the generated input value, performs its work and returns control patches. Programmatic generated-variable writes are reconciled at the end of the slot. If both a generated input value and its bridge property are changed, the changed generated value has the established precedence.

Direct Value assignment: read immediately in this same slot

```
form.dtAdmission.Value = "2024-02-29"
form.lblStatus.Text = form.dtAdmission.GetDate("dd-mm-YYYY")
```

A pending dtAdmission = "2024-02-29" assignment does not force an immediate form-proxy refresh earlier in that same handler. Use the direct Value form for same-slot formatting, or read GetDate in the next event. DatePicker method names are case-insensitive; the mask controls returned text only.

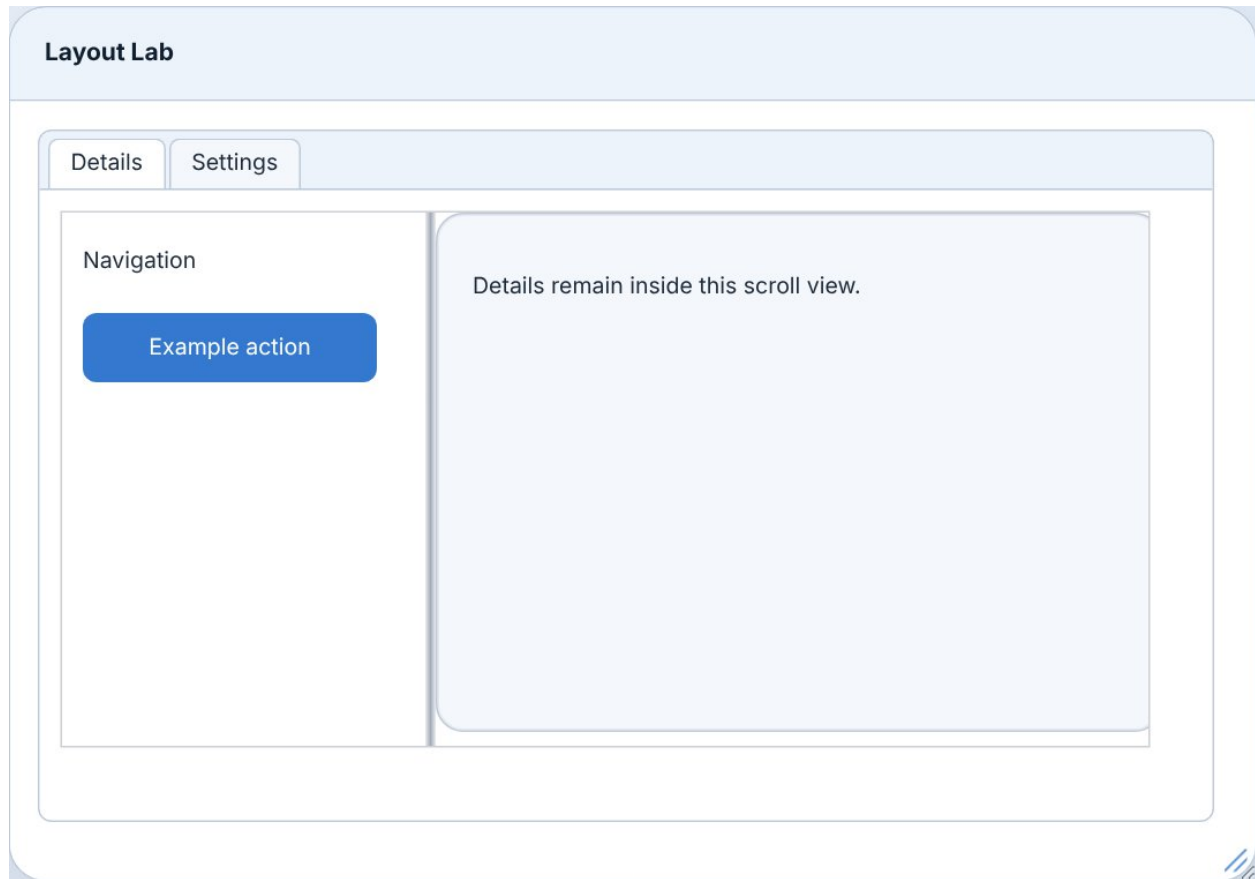
Closing and errors**Declare your own hasUnsavedChanges AS LOGICAL outside managed blocks**

```
PROCEDURE MainWindow_closing(sender, event, form)
  IF hasUnsavedChanges
    event.Cancel = .T.
  ENDIF
ENDPROC
```

Cancellation applies to an explicit form-close request. It is not guaranteed to prevent browser-tab closure, host shutdown or operating-system termination. Runtime errors should be caught where recovery is meaningful. The date bridge restores its invalid date subset, not every user variable or external side effect.

Layouts, themes and multiple forms

Start with correct parents and logical dimensions. Use Panel for grouping, TabView with TabPage for tabs, Splitter for adjustable panes, ScrollView for overflow and DockPanel for edge/fill arrangements. Layout is not inferred from visual overlap.



Retained layout example showing tabs, split panes and scrolling.

Setting	Design consequence
x, y, width, height	Logical geometry; child coordinates relate to their container.
dock	none, top, bottom, left, right or fill, where accepted by the schema.
sizeMode	fixed, responsive or fullscreen on supported form roots.
scaleContent / preserveAspectRatio	Presentation scaling is separate from the saved logical control positions.
allowUpscale	Controls whether a smaller design can be enlarged; test text readability.
theme	midnight, dark, aqua, light, ocean, forest, sunset or graphite.

Test a small and a large viewport. Check keyboard focus, readable text, tab content and scrollbars rather than only comparing outer form bounds. User dragging and programmatic Splitter.Position use different paths; the latter remains an open issue in this edition.

Separate windows and embedded subforms

Declare each form in kokum.toml and associate it with its slots module. A normal Window is a separate form; an embedded subform uses formKind=subform and a SubForm host in its parent. Preserve the existing UI instance for HIDE and recreate it after DISPOSE when shown again.

Manifest fragment; both referenced forms must exist

```
[[form]]
name = "MainWindow"
source = "ui/main_window.kform"
slots = "src/slots.kokum"

[[form]]
name = "CustomerView"
source = "ui/customer.kform"
slots = "src/slots.kokum"
```

Use separate actions/handlers, not all three in one demonstration click

```
SHOW CustomerView
HIDE CustomerView
DISPOSE CustomerView
```

Form UI lifetime and PUBLIC module state are not the same. A disposed form can be recreated while the owning module's variables remain. Avoid duplicate IDs across the forms sharing one slots module. ModalDialog is a retained schema entry with a known creation restriction; use a separately configured modal Window for a supported alternative, then test focus and closure explicitly. [S02, S03]

Practical labs: dates, data and background work

Lab 1 — admission date and display masks

Create dtAdmission and lblStatus, plus btnRead, btnClear and btnLeap. Set a permitted date range in the Inspector. Connect the three Button signals to the routines below; connect dateChanged to btnRead_clicked if the same formatting should run after a user selection.

Kokum example

```
EXPORT PROCEDURE btnRead_clicked
EXPORT PROCEDURE btnClear_clicked
EXPORT PROCEDURE btnLeap_clicked

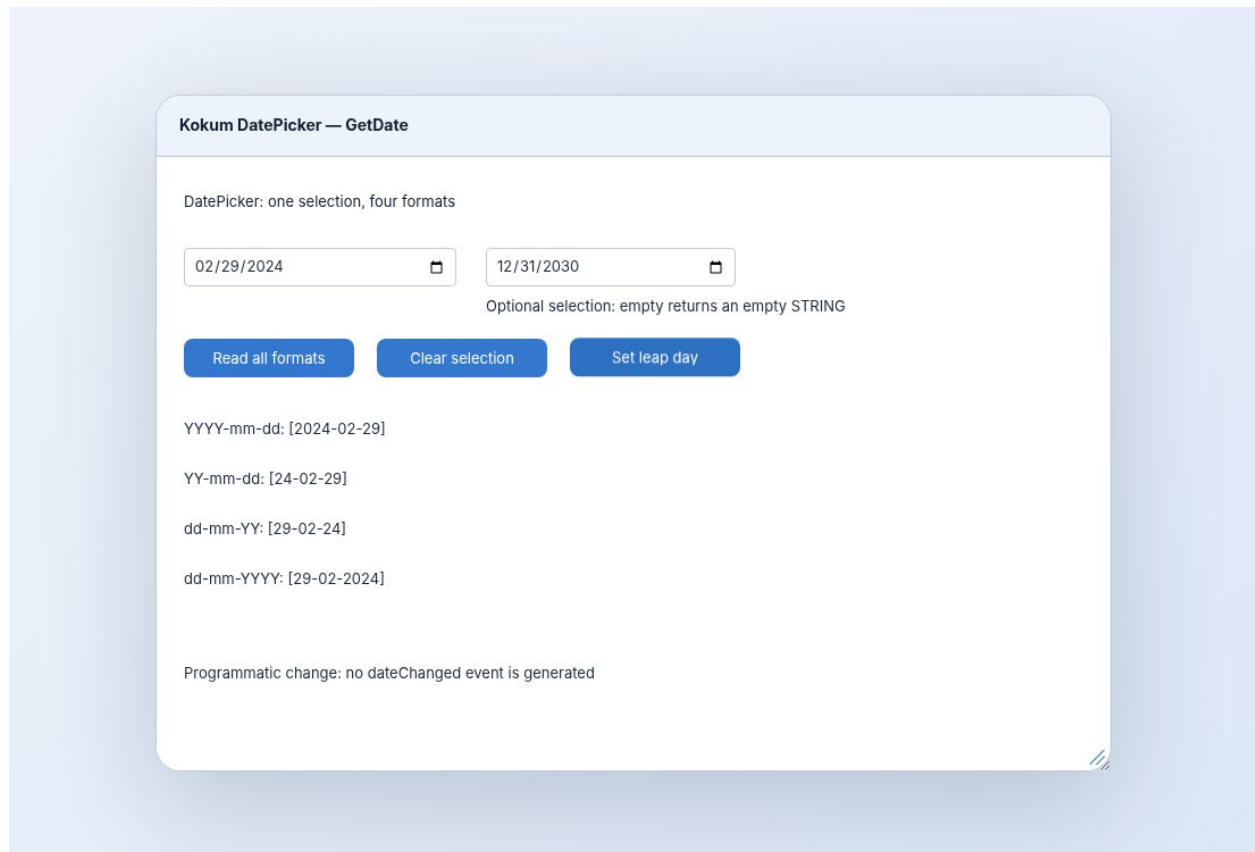
PROCEDURE btnRead_clicked(sender, event, form)
    form.lblStatus.Text = ;
    form.dtAdmission.GetDate("dd-mm-YYYY")
ENDPROC

PROCEDURE btnClear_clicked(sender, event, form)
    dtAdmission = ""
    form.lblStatus.Text = "No date selected"
ENDPROC

PROCEDURE btnLeap_clicked(sender, event, form)
    form.dtAdmission.MinDate = "2020-01-01"
    form.dtAdmission.MaxDate = "2030-12-31"
    form.dtAdmission.Value = "2024-02-29"
    form.lblStatus.Text = ;
    form.dtAdmission.GetDate("dd-mm-YYYY")
ENDPROC
```

Mask	For selected 2026-09-20
YYYY-mm-dd	2026-09-20
YY-mm-dd	26-09-20
dd-mm-YY	20-09-26
dd-mm-YYYY	20-09-2026

All masks contain one year, month and day token. Their token spelling is ASCII case-insensitive. No time, month-name or timezone formatting is implied. Empty selection returns an empty STRING. New DatePicker forms require GUI 1.9; older ordinary forms retain their compatibility requirements. [S13]



Current supplied DatePicker/GetDate example preview. The getter formats text without changing the calendar's localized display.

Lab 2 — a local database and a table

Add SQLite dbLocal, TableView tblPeople with designed columns id and name, a TextBox txtName, Buttons Create/Insert/Refresh and lblStatus. Use a private disposable database for this lab. The generated DATABASE component requires matching external configuration; its module name must equal your actual slots module.

db_env.conf — replace module/path to match the project

```
[database:dbLocal]
provider=sqlite
module=book.slots
file=/absolute/path/to/private-lab/people.sqlite3
auto_open=true
open_mode=readwrite-create
create_parent_directory=true
journal_mode=DELETE
synchronous=FULL
foreign_keys=true
busy_timeout_ms=1000
```

Linux terminal

```
export KOKUM_DB_ENV=/absolute/path/to/private-lab/db_env.conf
kokumvm /absolute/path/to/people.kapp
```

Connect/export these normal slots through Events

```
PROCEDURE btnCreate_clicked(sender, event, form)
    dbLocal.Execute( ;
        "CREATE TABLE IF NOT EXISTS people " + ;
        "(id INTEGER PRIMARY KEY, name TEXT)" )
    form.lblStatus.Text = "Ready"
ENDPROC

PROCEDURE btnInsert_clicked(sender, event, form)
    dbLocal.Execute( ;
        "INSERT INTO people(name) VALUES (?)", ;
        [TRIM(txtName)])
    tblPeople = dbLocal.QueryTable( ;
        "SELECT id,name FROM people ORDER BY id", ;
        NULL, .T.)
ENDPROC

PROCEDURE btnRefresh_clicked(sender, event, form)
    tblPeople = dbLocal.QueryTable( ;
        "SELECT id,name FROM people ORDER BY id", ;
        NULL, .T.)
ENDPROC
```

This lab uses a single automatic ARRAY owner for tblPeople. Do not follow the array assignment with a later native RowCount/SetCell handoff and expect the known defect to be repaired. Parameter arrays separate data from SQL; avoid concatenating txtName into the SQL statement. Use NULL deliberately when a field is missing. [S07]

Lab 3 — asynchronous work with owner-side polling

Add btnStart, timerPoll with Interval=100 and Running=false, and lblStatus. Keep a user-owned PUBLIC pending variable outside generated blocks. A task result is collected by the Timer slot; the worker does not retain or mutate form controls.

Connect/export both event procedures; initialize pending to NULL in your own startup

```
PUBLIC pending

ASYNC FUNCTION Calculate AS INTEGER
    SLEEP(250)
    RETURN 42
ENDFUNC

PROCEDURE btnStart_clicked(sender, event, form)
    IF pending != NULL
        IF NOT pending.IsDone
            RETURN
        ENDIF
    ENDIF
    pending = Calculate()
    form.timerPoll.Running = .T.
    form.lblStatus.Text = "Working"
ENDPROC

PROCEDURE timerPoll_timerTick(sender, event, form)
    IF pending = NULL
        RETURN
    ENDIF
    IF NOT pending.IsDone
        RETURN
    ENDIF
    form.timerPoll.Running = .F.
    IF pending.IsFailed
        form.lblStatus.Text = pending.ErrorMessage
    ELSE
        form.lblStatus.Text = STR(pending.Result())
    ENDIF
    pending = NULL
ENDPROC
```

Expected final label is 42. The GUI uses a persistent bytecode VM session. ASYNC semantics in the AST/IR reference paths are not proof of an independently running GUI thread. Repeated Result() reads the same cached result; timeout does not cancel arbitrary work. [S06]

Lab 4 — HTTP and WebSocket integration

For a URL request, put finite transport settings in the worker and return a plain MAP. For WebSocketClient, configure the native component while disconnected, establish a connection, then drain a bounded number of queued messages per Timer tick. Initial Connect remains a blocking call. Do not transfer a live WEBSOCKET into a TASK.

Timer slot body; wsFeed must be configured and connected

```
LOCAL message
LOCAL i AS INTEGER
FOR i = 1 TO 8
    message = wsFeed.Poll(0)
    IF message = NULL
        EXIT
    ENDIF
    form.lblStatus.Text = JSONENCODE(message)
NEXT
```

Use verified HTTPS/WSS for untrusted networks and keep tokens outside bundles. HTTP status failures are response maps, while transport failures are exceptions. The function reference includes the exact URL request/response fields and separate local-service setups. [S09, S10]

Widget reference A-Z

All 38 entries in the current executable schema are included. IDs are append-only and not necessarily contiguous. ModalDialog and SubFormHost are explicitly restricted/legacy entries; CustomCanvas is not part of this schema. A native property name does not prove that every browser behavior is complete.

Property tables list control-specific settings and source defaults. Common geometry, font/color, enabled and visible fields are described in the next chapter. The Inspector uses its own labels; runtime access commonly uses PascalCase through form.<id>. Collections stored as serialized property text may expose ARRAY/MAP values in slots.

Control	Category	Status
Button	Basic	Current schema
CheckBox	Input	Current schema
ComboBox	Data	Current schema
DatePicker	Input	Current schema
DockPanel	IDE Containers	Current schema
Graph	Data Visualization	Current schema
Image	Display	Current schema
Label	Basic	Current schema
ListBox	Data	Current schema
ListView	IDE Data	Current schema
Menu	Menus	Current schema
MenuBar	Menus	Current schema
MenuItem	Menus	Current schema
ModalDialog	IDE Containers	Legacy / restricted
Panel	Containers	Current schema
PopupMenu	IDE Commands	Current schema
PostgreSQL	Database	Current schema
ProgressBar	Data	Current schema
PropertyGrid	IDE Data	Current schema
RadioButton	Input	Current schema
ScrollView	IDE Containers	Current schema
Splitter	IDE Containers	Current schema
SQLite	Database	Current schema
StatusBar	Containers	Current schema
SubForm	Containers	Current schema
SubFormHost	Compatibility	Legacy / restricted

Control	Category	Status
TableView	IDE Data	Current schema
TabPage	IDE Containers	Current schema
TabView	IDE Containers	Current schema
TextArea	Input	Current schema
TextBox	Input	Current schema
TigerDB	Database	Current schema
Timer	Nonvisual	Current schema
ToolBar	IDE Commands	Current schema
ToolButton	IDE Commands	Current schema
TreeView	IDE Data	Current schema
WebSocketClient	Networking	Current schema
Window	Containers	Current schema

Button

Start an action through the clicked signal.

Set up in the Designer

Create btnGreet and lblStatus. Select btnGreet, open Events, create/connect btnGreet_clicked, then edit its generated procedure.

Default creation size: 90 × 30 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
default	LOGICAL	.F.	Default
checkable	LOGICAL	.F.	Checkable

Signals: clicked, doubleClicked, keyDown, keyUp, focusReceived, focusLost, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
form.lblStatus.Text = "Hello, Kokum"
```

Expected: Clicking the bound Button changes the label.

Boundary / practical note

Setting a caption does not create a handler. Keep network/database work bounded or delegate it to a supported worker.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

CheckBox

Represent an independent on/off choice.

Set up in the Designer

Add `chkReady` and `btnSave`. Use the `checkedChanged` signal to update the action state.

Default creation size: 120 × 24 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
checked	LOGICAL	.F.	Checked
group	STRING	""	Group

Signals: `clicked`, `keyDown`, `keyUp`, `focusReceived`, `focusLost`, `checkedChanged`, `contextMenu`, `mouseDown`, `mouseUp`

Small slot example — use the controls named above

```
form.btnSave.Enabled = chkReady
```

Expected: The Button becomes enabled when the checkbox is checked.

Boundary / practical note

The generated variable is LOGICAL. Multiple independent checkboxes do not need a radio group.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ComboBox

Choose a single string from an option list, with optional editable input.

Set up in the Designer

Create `cmbRole`. In Properties, open Items and enter one item per line. Set Selected Index or assign the generated value. The full native item API is in the function reference.

Default creation size: 160 × 28 logical units.

Property	Type	Default	Choice / meaning
editable	LOGICAL	.F.	Editable
selectedIndex	INTEGER	-1	Selected Index
placeholder	STRING	""	Placeholder
items	STRING	""	Items

Signals: `keyDown`, `keyUp`, `focusReceived`, `focusLost`, `selectionChanged`, `contextMenu`, `mouseDown`, `mouseUp`

Small slot example — use the controls named above

```
form.cmbRole.SetItems(["Analyst", "Developer"])
cmbRole = "Developer"
```

Expected: Two options are loaded and Developer is selected.

Boundary / practical note

`cmbRole` is the selected STRING; `form.cmbRole` is the control MAP. Item APIs use one-based insertion/removal positions; `SelectedIndex` is zero-based, with -1 meaning no selection.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

DatePicker

Select a calendar date and exchange a stable ISO date STRING with Kokum.

Set up in the Designer

Add dtAdmission from Input → Date Picker. Set Value, Min Date, Max Date and Read Only. Bind dateChanged when needed; an unbound date still synchronizes before another event.

Default creation size: 180 × 30 logical units.

Property	Type	Default	Choice / meaning
readOnly	LOGICAL	.F.	Read Only
value	STRING	""	Date
minDate	STRING	""	Minimum Date
maxDate	STRING	""	Maximum Date

Signals: dateChanged, enterPressed, keyDown, keyUp, focusReceived, focusLost, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
form.dtAdmission.Value = "2026-09-20"
form.lblStatus.Text = form.dtAdmission.GetDate("dd-mm-YYYY")
```

Expected: Displays 20-09-2026, while the stored value remains 2026-09-20.

Boundary / practical note

The bare dtAdmission variable is STRING. Call the method on form.dtAdmission. Empty selection returns ""; years 0001..9999 use calendar validation. New DatePicker forms require Linux GUI contract 1.9.

Source: [S02, S03] and [S13]. A snippet does not create the controls or its event binding.

DockPanel

Arrange child containers along available edges and a remaining center region.

Set up in the Designer

Create dockMain. Add panelNavigation with Dock=left and panelContent with Dock=fill. Check insertion order and size constraints.

Default creation size: 500 × 320 logical units.

Property	Type	Default	Choice / meaning
layout	STRING	'absolute'	absolute, horizontal, vertical, dock
spacing	INTEGER	4	Spacing
padding	INTEGER	4	Padding

Signals: No own event signals; use configuration and explicit calls.

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, ToolBar, DockPanel, PropertyGrid, SubFormHost, Graph, DatePicker.

Small slot example — use the controls named above

```
form.panelNavigation.Width = 220
```

Expected: Changes the navigation pane's logical width.

Boundary / practical note

DockPanel does not automatically fix incompatible constraints or arbitrary overlapping child coordinates. Test actual resized output.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Graph

Render line, bar, pie or candle data using a generated GRAPH resource.

Set up in the Designer

Create graphSales. Choose Graph Type, titles, palette and legend/grid options; use the generated graphSales handle for methods.

Default creation size: 480 × 280 logical units.

Property	Type	Default	Choice / meaning
graphType	STRING	'line'	line, bar, pie, candle
title	STRING	""	Title
showLegend	LOGICAL	.T.	Show Legend
showGrid	LOGICAL	.T.	Show Grid
animation	LOGICAL	.T.	Animate Updates
smooth	LOGICAL	.F.	Smooth Lines
stacked	LOGICAL	.F.	Stack Bars
lineWidth	NUMBER	2	Line Width
maxPoints	INTEGER	1000	Maximum Points
xAxisTitle	STRING	""	X Axis Title
yAxisTitle	STRING	""	Y Axis Title
palette	STRING	'theme'	theme, classic, cool, warm, market, monochrome
emptyText	STRING	'No data'	Empty Text

Signals: keyDown, keyUp, focusReceived, focusLost, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
graphSales.SetType("bar")
graphSales.SetTitle("Quarterly sales")
graphSales.SetData(["Q1", "Q2", "Q3"], [120, 150, 135], "Sales")
```

Expected: Supplies a three-point bar series.

Boundary / practical note

GRAPH is owner-session native state. For worker calculations, return plain arrays/maps and apply them in the owning slot. Do not transfer graphSales into a TASK.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Image

Display designed or dynamically loaded PNG/JPEG imagery.

Set up in the Designer

Create imgLogo. For a portable static image, add a resource with target web/images/kokum.png and set Source to images/kokum.png. Add Alternate Text.

Default creation size: 160 × 120 logical units.

Property	Type	Default	Choice / meaning
source	STRING	""	Source
scaleMode	STRING	'fit'	none, fit, fill, stretch, center
alternateText	STRING	""	Alternate Text

Signals: clicked, doubleClicked, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
? imgLogo.SetImage("images/kokum.png", .T.)
```

Expected: Loads the host-relative PNG and returns true when the native load succeeds.

Boundary / practical note

Dynamic SetImage/Clear refresh has an unresolved rendered-path issue. This example requires a real file in the runtime working directory; an arbitrary bundled path is not automatically a local filesystem file.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Label

Show a caption, calculated value, status message or instruction.

Set up in the Designer

Add a Label named lblStatus. Set Text, Alignment and Word Wrap; choose a readable foreground color.

Default creation size: 120 × 22 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
alignment	STRING	'left'	left, center, right
wordWrap	LOGICAL	.T.	Word Wrap

Signals: clicked, doubleClicked, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
form.lblStatus.Text = "Ready to save"
```

Expected: The existing label text is replaced.

Boundary / practical note

Labels do not generate a PUBLIC text variable. Address the control through form.lblStatus.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ListBox

Present a visible list with single or multiple selection.

Set up in the Designer

Create lstTopics, set Items and Multi Select, then connect selectionChanged when needed.

Default creation size: 180 × 120 logical units.

Property	Type	Default	Choice / meaning
selectedIndex	INTEGER	-1	Selected Index
multiSelect	LOGICAL	.F.	Multiple Selection
items	STRING	""	Items

Signals: doubleClicked, keyDown, keyUp, focusReceived, focusLost, selectionChanged, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
form.lstTopics.Items = ["Kokum", "SQL", "Testing"]
lstTopics = ["SQL", "Testing"]
```

Expected: Loads choices and selects the listed values when Multi Select is enabled.

Boundary / practical note

The automatic value is ARRAY even for a single selection. Do not assume ComboBox native methods are available on ListBox.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ListView

Present selectable items with configured list/view appearance.

Set up in the Designer

Create listRecords. Set Items, View Mode, Item Height and Multi Select; bind selectionChanged or itemActivated.

Default creation size: 260 × 220 logical units.

Property	Type	Default	Choice / meaning
selectedIndex	INTEGER	-1	Selected Index
multiSelect	LOGICAL	.F.	Multiple Selection
items	STRING	""	Items
viewMode	STRING	'list'	list, icons, details
itemHeight	INTEGER	22	Item Height

Signals: doubleClicked, keyDown, keyUp, focusReceived, focusLost, selectionChanged, itemActivated, contextMenu, mouseDown, mouseUp, selectionChanging

Small slot example — use the controls named above

```
form.listRecords.Items = ["First", "Second"]
form.listRecords.SelectedIndex = 1
```

Expected: Loads two items and selects the second.

Boundary / practical note

The automatic value is ARRAY. Native calls must come from the supported property surface, not guessed ListBox/TableView methods.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Menu

Group related menu items under a top-level caption.

Set up in the Designer

Insert Menu beneath MenuBar, set Text to File, then add MenuItem children such as Save and Exit.

Default creation size: 140 × 24 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
shortcut	STRING	""	Shortcut

Signals: No own event signals; use configuration and explicit calls.

Allowed children: MenuItem.

Small slot example — use the controls named above

```
form.lblStatus.Text = "File action"
```

Expected: A leaf MenuItem calls this action body.

Boundary / practical note

Do not attach a Menu directly where only a MenuItem is allowed. Nested command levels use MenuItem children.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

MENUS / SCHEMA ID 11

MenuBar

Hold top-level menus such as File, Edit and Help.

Set up in the Designer

Add Menu Bar directly to the root Window. Add Menu children; add MenuItem children below a Menu. Use the hierarchy-aware menu editor.

Default creation size: 640 × 24 logical units.

Signals: No own event signals; use configuration and explicit calls.

Allowed children: Menu.

Small slot example — use the controls named above

```
form.lblStatus.Text = "Menu command received"
```

Expected: This body can be used in a MenuItem's menuSelected slot.

Boundary / practical note

MenuBar is structural and has no own menuSelected signal. Attach actions to the leaf MenuItem.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

MenuItem

Execute a command, toggle a checked choice or provide a separator.

Set up in the Designer

Create menuSave under a Menu and connect menuSelected. Set Text, Shortcut and Enabled. For a separator, set Separator rather than an arbitrary caption.

Default creation size: 160 × 24 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
checked	LOGICAL	.F.	Checked
shortcut	STRING	""	Shortcut
separator	LOGICAL	.F.	Separator
command	STRING	""	Command

Signals: menuSelected, contextMenu, command, mouseDown, mouseUp

Allowed children: MenuItem.

Small slot example — use the controls named above

```
form.lblStatus.Text = "Save selected"
```

Expected: The bound menu command updates the status.

Boundary / practical note

Enabled=false should prevent activation. Treat menuSelected as the action signal; do not wire duplicate signals to the same side effect without a reason.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ModalDialog

A schema-listed dialog container retained for compatibility, not a validated new-design path in this release.

Set up in the Designer

Do not choose it as a new root or arbitrary child: the recorded compiler checks rejected both. Use a separately declared Window with modal=true and test that path.

Default creation size: 480 × 320 logical units.

Property	Type	Default	Choice / meaning
title	STRING	'Kokum Application'	Title
minWidth	INTEGER	0	Legacy Minimum Width
minHeight	INTEGER	0	Legacy Minimum Height
maxWidth	INTEGER	0	Legacy Maximum Width
maxHeight	INTEGER	0	Legacy Maximum Height
resizable	LOGICAL	.T.	Resizable
sizeMode	STRING	'responsive'	fixed, responsive, fullscreen
viewportWidth	NUMBER	90	Viewport Width (%)
viewportHeight	NUMBER	86	Viewport Height (%)
scaleContent	LOGICAL	.T.	Scale Absolute Content
preserveAspectRatio	LOGICAL	.T.	Preserve Aspect Ratio
allowUpscale	LOGICAL	.F.	Allow Runtime Upscale
theme	STRING	'midnight'	midnight, dark, aqua, light, ocean, forest, sunset, graphite
modal	LOGICAL	.F.	Modal
layout	STRING	'absolute'	absolute, horizontal, vertical, dock
spacing	INTEGER	4	Spacing
padding	INTEGER	4	Padding

Signals: loaded, closing, closed

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, Toolbar, DockPanel, PropertyGrid, SubFormHost, Graph, DatePicker.

Small slot example — use the controls named above

```
SHOW CustomerDialog
```

Expected: Shows a separately compiled Window named CustomerDialog, not an unsupported ModalDialog-root form.

Boundary / practical note

The example is explicitly the supported Window alternative. ModalDialog has no claimed working root/child example in this baseline.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Panel

Group related controls inside one layout container. Coordinates of its children are relative to the panel.

Set up in the Designer

Drop Panel on the Window, then place Label and TextBox inside it. Use Hierarchy to confirm they are children, not merely visually overlapping siblings.

Default creation size: 240 × 180 logical units.

Property	Type	Default	Choice / meaning
layout	STRING	'absolute'	absolute, horizontal, vertical, dock
spacing	INTEGER	4	Spacing
padding	INTEGER	4	Padding

Signals: No own event signals; use configuration and explicit calls.

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, ToolBar, DockPanel, PropertyGrid, SubFormHost, Graph, DatePicker.

Small slot example — use the controls named above

```
form.panelDetails.X = 24
form.panelDetails.Y = 80
```

Expected: Moves the panel without changing the children's parent-relative coordinates.

Boundary / practical note

A panel is not a separate form or VM session. It does not acquire independent loaded/closing slots.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

PopupMenu

Define a nonvisual context-menu tree with MenuItem children.

Set up in the Designer

Insert PopupMenu on a compatible root and add MenuItem children. Keep its command handlers separate from the event that requests a context menu.

Default creation size: 0 × 0 logical units.

Signals: No own event signals; use configuration and explicit calls.

Allowed children: MenuItem.

Small slot example — use the controls named above

```
form.lblStatus.Text = "Context command selected"
```

Expected: This is a valid MenuItem command body once its menuSelected binding is invoked.

Boundary / practical note

The schema and native menu callback are present; end-to-end browser popup activation was not certified in the retained acceptance record. No unsupported ShowPopup method is invented here.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

PostgreSQL

Use libpq-backed PostgreSQL queries from a nonvisual DATABASE component.

Set up in the Designer

Create dbPg and an external configuration matching its id/module. Supply a permitted database/user and configure sslmode deliberately.

Property	Type	Default	Choice / meaning
host	STRING	'127.0.0.1'	Host
port	INTEGER	5432	Port
database	STRING	'postgres'	Database
username	STRING	'appuser'	Username
password	STRING	'\${ENV:KOKUM_DBPOSTGRES_PASSWORD}'	Password / Environment Reference
schema	STRING	'public'	Schema
sslMode	STRING	'prefer'	disable, allow, prefer, require, verify-ca, verify-full
sslRootCert	STRING	''	Root CA File
sslCert	STRING	''	Client Certificate
sslKey	STRING	''	Client Key
applicationName	STRING	'Kokum'	Application Name
autoOpen	LOGICAL	.T.	Auto Connect
connectTimeout	INTEGER	10000	Connect Timeout (ms)
queryTimeout	INTEGER	30000	Query Timeout (ms)
readOnly	LOGICAL	.F.	Read Only

Signals: No own event signals; use configuration and explicit calls.

Small slot example — use the controls named above

```

LOCAL row
row = dbPg.QueryOne("SELECT name FROM people WHERE id = $1", [1])
IF row != NULL
    form.lblStatus.Text = STR(row["name"])
ENDIF

```

Expected: Displays the selected name when the configured server has that record.

Boundary / practical note

This is an operator-provided server example. It does not install PostgreSQL, migrate production tables or prove all remote authentication/TLS configurations.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ProgressBar

Show numeric progress within a configured inclusive range.

Set up in the Designer

Add progressWork. Set Minimum=0, Maximum=100 and initial Value=0. Use Indeterminate only when a numerical percentage is unavailable.

Default creation size: 180 × 20 logical units.

Property	Type	Default	Choice / meaning
minimum	NUMBER	0	Minimum
maximum	NUMBER	100	Maximum
value	NUMBER	0	Value
indeterminate	LOGICAL	.F.	Indeterminate

Signals: No own event signals; use configuration and explicit calls.

Small slot example — use the controls named above

```
form.progressWork.Minimum = 0
form.progressWork.Maximum = 100
form.progressWork.Value = 40
```

Expected: Displays progress at 40 in the configured range.

Boundary / practical note

A progress bar does not run work in the background. Update it from the owning slot/Timer after receiving worker results.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

PropertyGrid

Display editable named values grouped into categories.

Set up in the Designer

Create gridInfo. Populate through native property methods or the Properties editor. Give properties unique names and specify field type/readOnly deliberately.

Default creation size: 280 × 360 logical units.

Property	Type	Default	Choice / meaning
readOnly	LOGICAL	.F.	Read Only
items	STRING	""	Items
editable	LOGICAL	.F.	Editable
selectedProperty	STRING	""	Selected Property
categoryMode	LOGICAL	.T.	Category Mode
properties	STRING	""	Properties

Signals: keyDown, keyUp, focusReceived, focusLost, selectionChanged, itemActivated, contextMenu, mouseDown, mouseUp, selectionChanging

Small slot example — use the controls named above

```
form.gridInfo.Clear()
form.gridInfo.AddProperty( ;
    "General", "title", "Title", "Customer view", "string")
? form.gridInfo.GetValue("title")
```

Expected: Adds and reads the title value.

Boundary / practical note

gridInfo is a control MAP through form, not a generated DATABASE-style resource. Per-field read-only is different from whole-control Enabled.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

RadioButton

Choose one option within a mutually exclusive group.

Set up in the Designer

Create radBasic and radExpert. Set the same Group value on both and choose one initial Checked state.

Default creation size: 120 × 24 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
checked	LOGICAL	.F.	Checked
group	STRING	""	Group

Signals: clicked, keyDown, keyUp, focusReceived, focusLost, checkedChanged, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
IF radExpert
    form.lblStatus.Text = "Expert selected"
ELSE
    form.lblStatus.Text = "Basic selected"
ENDIF
```

Expected: Reads the generated LOGICAL value for the selected choice.

Boundary / practical note

Use a consistent group or parent grouping. Do not manually force conflicting checked states across a group.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ScrollView

Clip oversized child content and expose scrollbars.

Set up in the Designer

Create scrollDetails; place content inside it and choose horizontal/vertical scrollbar policies. Make the content larger than the viewport to test.

Default creation size: 320 × 240 logical units.

Property	Type	Default	Choice / meaning
scrollX	INTEGER	0	Scroll X
scrollY	INTEGER	0	Scroll Y
horizontalScrollBar	LOGICAL	.T.	Horizontal Scroll Bar
verticalScrollBar	LOGICAL	.T.	Vertical Scroll Bar

Signals: keyDown, keyUp, focusReceived, focusLost, scrollChanged, contextMenu, mouseDown, mouseUp

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, ToolBar, DockPanel, PropertyGrid, SubFormHost, Graph, DatePicker.

Small slot example — use the controls named above

```
form.scrollDetails.ScrollY = 60
```

Expected: Sets the requested vertical scroll position within available content.

Boundary / practical note

A scrollbar cannot scroll when no content overflows. Verify child parenting before changing scroll values.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Splitter

Let the user resize adjacent panes.

Set up in the Designer

Place two container children under Splitter. Set Orientation, Position and Minimum Pane Size, then test dragging in Run.

Default creation size: 500 × 300 logical units.

Property	Type	Default	Choice / meaning
orientation	STRING	'horizontal'	horizontal, vertical
position	INTEGER	200	Position
minimumPaneSize	INTEGER	40	Minimum Pane Size

Signals: keyDown, keyUp, focusReceived, focusLost, splitterMoved, contextMenu, mouseDown, mouseUp, mouseMove

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, ToolBar, DockPanel, PropertyGrid, SubFormHost, Graph, DatePicker.

Small slot example — use the controls named above

```
form.lblStatus.Text = "Divider moved"
```

Expected: Use this body in splitterMoved to observe the user interaction.

Boundary / practical note

Known issue: a native Position patch can be accepted but remain visually unapplied. User dragging and programmatic positioning are separate paths; do not treat the latter as fixed.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

SQLite

Connect a nonvisual DATABASE component to a local SQLite file.

Set up in the Designer

Place SQLite as dbLocal. Create an external private db_env.conf matching the component id and slots module, set KOKUM_DB_ENV, and open the connection or enable auto_open.

Property	Type	Default	Choice / meaning
databaseFile	STRING	'\${USER_DATA_DIR}/application.sqlite3'	Database File
openMode	STRING	'readwrite-create'	readwrite-create, readwrite, readonly
autoOpen	LOGICAL	.T.	Auto Open
createParentDirectory	LOGICAL	.T.	Create Parent Directory
busyTimeout	INTEGER	5000	Busy Timeout (ms)
journalMode	STRING	'WAL'	DELETE, TRUNCATE, PERSIST, MEMORY, WAL, OFF
foreignKeys	LOGICAL	.T.	Foreign Keys
synchronous	STRING	'NORMAL'	OFF, NORMAL, FULL, EXTRA
initializationScript	STRING	""	Initialization Script

Signals: No own event signals; use configuration and explicit calls.

Small slot example — use the controls named above

```
LOCAL count
count = dbLocal.QueryValue("SELECT COUNT(*) FROM people")
form.lblStatus.Text = STR(count)
```

Expected: Displays the number of rows after the SQLite setup lab has created people.

Boundary / practical note

The component is configuration-backed; PUBLIC dbLocal AS DATABASE alone does not open a database. Use bound SQL parameters and explicit transactions for updates.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

StatusBar

Display concise application progress or context along an edge.

Set up in the Designer

Add statusMain, set Dock to bottom and give it readable Text.

Default creation size: 640 × 24 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text

Signals: No own event signals; use configuration and explicit calls.

Small slot example — use the controls named above

```
form.statusMain.Text = "All changes saved"
```

Expected: Replaces the status-bar text.

Boundary / practical note

StatusBar has no automatic STRING variable or periodic refresh of its own.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

SubForm

Create an inline, independently shown/hidden form surface within a GUI application.

Set up in the Designer

Drag SubForm into the Designer's authoring area beside MainWindow, name it CustomerPane and put its controls inside it. Runtime x/y are separate from designerX/designerY.

Default creation size: 520 × 360 logical units.

Property	Type	Default	Choice / meaning
title	STRING	'Kokum Application'	Title
designerX	INTEGER	900	Designer X
designerY	INTEGER	40	Designer Y
closable	LOGICAL	.T.	Closable
minWidth	INTEGER	0	Legacy Minimum Width
minHeight	INTEGER	0	Legacy Minimum Height
maxWidth	INTEGER	0	Legacy Maximum Width
maxHeight	INTEGER	0	Legacy Maximum Height
resizable	LOGICAL	.T.	Resizable
sizeMode	STRING	'responsive'	fixed, responsive, fullscreen
viewportWidth	NUMBER	90	Viewport Width (%)
viewportHeight	NUMBER	86	Viewport Height (%)
scaleContent	LOGICAL	.T.	Scale Absolute Content
preserveAspectRatio	LOGICAL	.T.	Preserve Aspect Ratio
theme	STRING	'midnight'	midnight, dark, aqua, light, ocean, forest, sunset, graphite
modal	LOGICAL	.F.	Modal
layout	STRING	'absolute'	absolute, horizontal, vertical, dock
spacing	INTEGER	4	Spacing
padding	INTEGER	4	Padding

Signals: loaded, closing, closed

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, MenuBar, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, ToolBar, DockPanel, PropertyGrid, PopupMenu, SubFormHost, SQLite, TigerDB, Graph, WebSocketClient, PostgreSQL, DatePicker.

Small slot example — use the controls named above

```
SHOW CustomerPane
HIDE CustomerPane
SHOW CustomerPane
```

Expected: Shows, hides and shows the pane while retaining its applicable state.

Boundary / practical note

DISPOSE closes the UI instance; module PUBLIC values can survive until the owning session ends. A later SHOW recreates the surface. Child ids must remain unique across the shared slots module.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

SubFormHost

Legacy placeholder for an iframe-hosted, separately declared subform.

Set up in the Designer

Open an existing project using SubFormHost. Its child form remains a Window with formKind=subform and hostControl naming the host; register both forms in the manifest.

Default creation size: 520 × 360 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text

Signals: No own event signals; use configuration and explicit calls.

Small slot example — use the controls named above

```
SHOW CustomerView
HIDE CustomerView
```

Expected: Requests showing and hiding an existing registered legacy view.

Boundary / practical note

Hidden from the new toolbox. Prefer inline SubForm for new projects. This is compatibility documentation, not a recommendation to start new forms with the legacy host.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

TableView

Display structured rows and columns, with selection and a native row/cell API.

Set up in the Designer

Create tblData with columns named id and name. Choose one data ownership model: an automatic ARRAY, or consistently method-owned rows.

Default creation size: 420 × 240 logical units.

Property	Type	Default	Choice / meaning
multiSelect	LOGICAL	.F.	Multiple Selection
columns	STRING	""	Columns
rows	STRING	""	Rows
selectedRow	INTEGER	-1	Selected Row
selectedColumn	INTEGER	-1	Selected Column
headersVisible	LOGICAL	.T.	Headers Visible
gridVisible	LOGICAL	.T.	Grid Visible

Signals: doubleClicked, keyDown, keyUp, focusReceived, focusLost, selectionChanged, itemActivated, columnResized, contextMenu, mouseDown, mouseUp, selectionChanging

Small slot example — use the controls named above

```
tblData = [ ;
    ["id", "name"], ;
    ["1", "Anil"], ;
    ["2", "Sara"] ;
]
```

Expected: Displays two rows with the matching designed header names.

Boundary / practical note

Known issue: after assigning an automatic ARRAY, later RowCount/SetCell may address stale native rows. Do not mix ownership styles. Native row/column method indices are zero-based; Kokum array indices are one-based.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

TabPage

Contain the controls belonging to one TabView page.

Set up in the Designer

Insert only beneath TabView. Set Text to the tab caption and add normal controls inside the page.

Default creation size: 400 × 268 logical units.

Property	Type	Default	Choice / meaning
layout	STRING	'absolute'	absolute, horizontal, vertical, dock
spacing	INTEGER	4	Spacing
padding	INTEGER	4	Padding
text	STRING	""	Text
source	STRING	""	Source
icon	STRING	""	Icon

Signals: No own event signals; use configuration and explicit calls.

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, ToolBar, DockPanel, PropertyGrid, SubFormHost, Graph, DatePicker.

Small slot example — use the controls named above

```
form.lblDetails.Text = "Details refreshed"
```

Expected: Updates a Label that is a child of the Details page.

Boundary / practical note

The label id remains unique in the shared slots module; placing it on another page does not create a new variable scope.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

TabView

Organize controls into selectable pages.

Set up in the Designer

Drop TabView and add TabPage children. Place page content under the corresponding TabPage. Set Selected Index and Tab Position.

Default creation size: 420 × 300 logical units.

Property	Type	Default	Choice / meaning
selectedIndex	INTEGER	-1	Selected Index
tabPosition	STRING	'top'	top, bottom, left, right
closable	LOGICAL	.F.	Closable Tabs

Signals: keyDown, keyUp, focusReceived, focusLost, tabChanged, contextMenu, mouseDown, mouseUp, selectionChanging

Allowed children: TabPage.

Small slot example — use the controls named above

```
form.tabsMain.SelectedIndex = 1
```

Expected: Selects the second tab when it exists.

Boundary / practical note

SelectedIndex is zero-based. A visible tab label does not mean its content is correctly parented; check Hierarchy.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

TextArea

Edit multiline text such as comments, addresses or a response body.

Set up in the Designer

Add txtNotes. Set Word Wrap, Read Only and Max Length as needed. Give it sufficient height and bind `textChanged` only when necessary.

Default creation size: 260 × 110 logical units.

Property	Type	Default	Choice / meaning
<code>wordWrap</code>	LOGICAL	<code>.T.</code>	Word Wrap
<code>text</code>	STRING	<code>""</code>	Text
<code>placeholder</code>	STRING	<code>""</code>	Placeholder
<code>readOnly</code>	LOGICAL	<code>.F.</code>	Read Only
<code>maxLength</code>	INTEGER	<code>0</code>	Maximum Length

Signals: `textChanged`, `enterPressed`, `keyDown`, `keyUp`, `focusReceived`, `focusLost`, `contextMenu`, `mouseDown`, `mouseUp`

Small slot example — use the controls named above

```
txtNotes = "First line\nSecond line"
```

Expected: The generated STRING and control display contain two lines.

Boundary / practical note

The two-line text is stored with a newline. Avoid a slow handler on every keystroke; changes may be debounced.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

TextBox

Edit a single-line string, including optional password masking.

Set up in the Designer

Add txtName and lblStatus. Use Placeholder for guidance rather than storing a prompt in Text. Bind enterPressed or a Button to a validation slot.

Default creation size: 180 × 28 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
placeholder	STRING	""	Placeholder
readOnly	LOGICAL	.F.	Read Only
maxLength	INTEGER	0	Maximum Length
password	LOGICAL	.F.	Password

Signals: textChanged, enterPressed, keyDown, keyUp, focusReceived, focusLost, contextMenu, mouseDown, mouseUp

Small slot example — use the controls named above

```
txtName = TRIM(txtName)
form.lblStatus.Text = "Name: " + txtName
```

Expected: Trims the selected input value and displays it.

Boundary / practical note

The tested Visible=false path remains a known shared-renderer issue. Password masks display only; it is not encryption. The generated variable is STRING.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

TigerDB

Use Kokum's native client through a DATABASE component on the application host.

Set up in the Designer

Create dbTiger and a matching external provider configuration. Use a dedicated test server/database and explicit verified TLS settings where required.

Property	Type	Default	Choice / meaning
endpoint	STRING	'server'	server, router
host	STRING	'127.0.0.1'	Host
port	INTEGER	9192	Port
database	STRING	'default'	Database
username	STRING	'appuser'	Username
password	STRING	'\${ENV:KOKUM_TIGERDB_PASSWORD}'	Password / Environment Reference
tls	LOGICAL	.T.	TLS
verifyTLS	LOGICAL	.T.	Verify TLS Certificate
caFile	STRING	""	CA File
tlsServerName	STRING	""	TLS Server Name
clientCertificate	STRING	""	Client Certificate
clientKey	STRING	""	Client Key
autoOpen	LOGICAL	.T.	Auto Connect
connectTimeout	INTEGER	10000	Connect Timeout (ms)
queryTimeout	INTEGER	30000	Query Timeout (ms)
maxMessageBytes	INTEGER	8388608	Maximum Message Bytes

Signals: No own event signals; use configuration and explicit calls.

Small slot example — use the controls named above

```
LOCAL rows AS ARRAY
rows = dbTiger.Query("SELECT id, name FROM people ORDER BY id")
form.lblStatus.Text = STR(LEN(rows))
```

Expected: Displays the row count when the configured test database contains people.

Boundary / practical note

Requires an actual compatible TigerDB server and credentials. No live server acceptance is claimed by this book; the client API is not the TigerDB server implementation.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Timer

Schedule periodic events without displaying a control.

Set up in the Designer

Drop Timer into the component tray as timerPoll. Set Interval in milliseconds, Running and Repeat; bind timerTick.

Property	Type	Default	Choice / meaning
interval	INTEGER	1000	Interval
running	LOGICAL	.F.	Running
repeat	LOGICAL	.T.	Repeat

Signals: timerTick

Small slot example — use the controls named above

```
form.timerPoll.Interval = 250
form.timerPoll.Running = .T.
```

Expected: Enables periodic owner-session polling.

Boundary / practical note

A Timer is not a worker thread or precise real-time scheduler. Keep each timerTick slot short and stop polling after a completed job.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ToolBar

Group compact command controls in a strip.

Set up in the Designer

Add a ToolBar; insert ToolButton, Label, TextBox or ComboBox children allowed by the schema. Set Orientation and Icon Size.

Default creation size: 640 × 32 logical units.

Property	Type	Default	Choice / meaning
layout	STRING	'absolute'	absolute, horizontal, vertical, dock
spacing	INTEGER	4	Spacing
padding	INTEGER	4	Padding
orientation	STRING	'horizontal'	horizontal, vertical
movable	LOGICAL	.T.	Movable
iconSize	INTEGER	16	Icon Size

Signals: No own event signals; use configuration and explicit calls.

Allowed children: Label, TextBox, ComboBox, ToolButton.

Small slot example — use the controls named above

```
form.lblStatus.Text = "Toolbar action"
```

Expected: Use this body in the child ToolButton's clicked slot.

Boundary / practical note

ToolBar is a container, not a command by itself. Add action handlers to its interactive children.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

ToolButton

Provide a compact Button-style command, optionally with an icon or checked state.

Set up in the Designer

Create toolRefresh under a ToolBar, assign Text/Source and Tooltip, then connect clicked.

Default creation size: 32 × 28 logical units.

Property	Type	Default	Choice / meaning
text	STRING	""	Text
default	LOGICAL	.F.	Default
checkable	LOGICAL	.F.	Checkable
checked	LOGICAL	.F.	Checked
shortcut	STRING	""	Shortcut
command	STRING	""	Command
source	STRING	""	Source

Signals: clicked, keyDown, keyUp, focusReceived, focusLost, checkedChanged, contextMenu, command, mouseDown, mouseUp

Small slot example — use the controls named above

```
form.lblStatus.Text = "Refresh requested"
```

Expected: Clicking the tool button executes the bound action.

Boundary / practical note

The image resource must be packaged separately. A Command label is not a substitute for a bound Kokum handler.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

TreeView

Display indented parent/child data and selection paths.

Set up in the Designer

Create treeProjects. In the Items editor, preserve indentation for child nodes. Choose Show Root and Multi Select as appropriate.

Default creation size: 220 × 300 logical units.

Property	Type	Default	Choice / meaning
multiSelect	LOGICAL	.F.	Multiple Selection
items	STRING	""	Items
selectedPath	STRING	""	Selected Path
showRoot	LOGICAL	.T.	Show Root
editable	LOGICAL	.F.	Editable

Signals: doubleClicked, keyDown, keyUp, focusReceived, focusLost, selectionChanged, itemActivated, itemExpanded, itemCollapsed, contextMenu, mouseDown, mouseUp, selectionChanging

Small slot example — use the controls named above

```
form.treeProjects.Items = ["Root", "  Child A", "  Child B"]
form.treeProjects.SelectedPath = "Root/Child A"
```

Expected: Builds a small hierarchy and selects Child A.

Boundary / practical note

Use SelectedPath, not an invented SelectedIndex. Browser-only convenience functions are not automatically callable from Kokum slots.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

WebSocketClient

Connect to an outbound ws/wss service and queue incoming data for the application.

Set up in the Designer

Drop WebSocket Client as wsFeed. Configure its endpoint and TLS options before connecting. Add Timer polling for incoming messages.

Property	Type	Default	Choice / meaning
url	STRING	'ws://127.0.0.1:8080/'	URL
autoConnect	LOGICAL	.F.	Auto Connect
subprotocols	STRING	""	Subprotocols
verifyTLS	LOGICAL	.T.	Verify TLS
caFile	STRING	""	CA File
tlsServerName	STRING	""	TLS Server Name
clientCertificate	STRING	""	Client Certificate
clientKey	STRING	""	Client Key
connectTimeout	INTEGER	10000	Connect Timeout (ms)
ioTimeout	INTEGER	30000	I/O Timeout (ms)
pingInterval	INTEGER	0	Ping Interval (ms)
maxMessageBytes	INTEGER	8388608	Maximum Message Bytes
queueLimit	INTEGER	1024	Queue Limit
reconnect	LOGICAL	.F.	Reconnect
reconnectDelay	INTEGER	1000	Reconnect Delay (ms)
maxReconnectAttempts	INTEGER	0	Maximum Reconnect Attempts

Signals: No own event signals; use configuration and explicit calls.

Small slot example — use the controls named above

```

LOCAL message
message = wsFeed.Receive(0)
IF message != NULL
    form.lblStatus.Text = JSONENCODE(message)
ENDIF

```

Expected: Reads one queued event without an unbounded wait.

Boundary / practical note

Connect is blocking; polling is not a background GUI callback. Use verified wss for credentials. The Windows-local networking fixes are not automatically adopted by this Linux source.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Window

The root application surface. It owns the control hierarchy, theme, startup and close lifecycle.

Set up in the Designer

Create a GUI Designer Project, select MainWindow in Hierarchy, then set Title, Width, Height and theme. Use one root Window per .kform; add other windows as separate forms.

Default creation size: 800 × 600 logical units.

Property	Type	Default	Choice / meaning
title	STRING	'Kokum Application'	Title
formKind	STRING	'window'	window, subform
hostControl	STRING	""	Subform Host
minWidth	INTEGER	0	Legacy Minimum Width
minHeight	INTEGER	0	Legacy Minimum Height
maxWidth	INTEGER	0	Legacy Maximum Width
maxHeight	INTEGER	0	Legacy Maximum Height
resizable	LOGICAL	.T.	Resizable
sizeMode	STRING	'responsive'	fixed, responsive, fullscreen
viewportWidth	NUMBER	90	Viewport Width (%)
viewportHeight	NUMBER	86	Viewport Height (%)
scaleContent	LOGICAL	.T.	Scale Absolute Content
preserveAspectRatio	LOGICAL	.T.	Preserve Aspect Ratio
allowUpscale	LOGICAL	.F.	Allow Runtime Upscale
theme	STRING	'midnight'	midnight, dark, aqua, light, ocean, forest, sunset, graphite
modal	LOGICAL	.F.	Modal
layout	STRING	'absolute'	absolute, horizontal, vertical, dock
spacing	INTEGER	4	Spacing
padding	INTEGER	4	Padding

Signals: loaded, closing, closed

Allowed children: Panel, Label, Button, TextBox, TextArea, CheckBox, RadioButton, ComboBox, ListBox, MenuBar, StatusBar, ProgressBar, Image, Timer, TabView, TreeView, ListView, TableView, Splitter, ScrollView, ToolBar, DockPanel, PropertyGrid, PopupMenu, SubFormHost, SubForm, SQLite, TigerDB, Graph, WebSocketClient, PostgreSQL, DatePicker.

Small slot example — use the controls named above

```
form.CloseRequested = .T.
```

Expected: Requests closure through the existing closing/closed lifecycle.

Boundary / practical note

Use a closing slot with event.Cancel only for explicit form closure. It cannot prevent an arbitrary browser tab or process from being terminated.

Source: [S02, S03]. A snippet does not create the controls or its event binding.

Common properties and event directory

These are source-schema names, not independent global functions. Not every component carries every common property: nonvisual database/network components mainly expose their own configuration. An asterisk is not used to imply unsupported inheritance; consult the relevant widget schema when a property is absent.

Schema property	Type	Common default	Meaning / options
x	INTEGER	0	X
y	INTEGER	0	Y
width	INTEGER	100	Width; actual creation dimensions are widget-specific.
height	INTEGER	28	Height; actual creation dimensions are widget-specific.
visible	LOGICAL	.T.	Visible
enabled	LOGICAL	.T.	Enabled
tabIndex	INTEGER	0	Tab Index
tooltip	STRING	"	Tool Tip
accessibleName	STRING	"	Accessible Name
cursor	STRING	'default'	default, arrow, text, crosshair, hand, resizeHorizontal, resizeVertical, move, wait
fontFamily	STRING	'system'	Font Family
fontSize	NUMBER	13	Font Size
fontWeight	INTEGER	400	Font Weight
textColor	STRING	'system'	Foreground Color
backgroundColor	STRING	'system'	Background Color
dock	STRING	'none'	none, left, right, top, bottom, fill
minimumWidth	INTEGER	0	Minimum Width
minimumHeight	INTEGER	0	Minimum Height
maximumWidth	INTEGER	0	Maximum Width
maximumHeight	INTEGER	0	Maximum Height

Event selection

Use the exact signal advertised for the selected type. clicked belongs to Buttons/ToolButtons; menuSelected belongs to MenuItem; selectionChanged is distinct from selectionChanging; textChanged and checkedChanged describe input changes. Window lifecycle signals are loaded, closing and closed. Timer uses timerTick. DatePicker adds dateChanged.

keyDown/keyUp, focusReceived/focusLost, contextMenu, mouseDown/mouseUp and itemActivated appear only where the source schema lists them. Returning from a handler does not automatically reverse changes. Native event payload fields vary by event; do not assume every event contains a selected row or a date.

Input-value mapping

The displayed property and automatic value may have different names: TextBox uses Text, ComboBox exposes SelectedText, CheckBox uses Checked, TableView uses Rows and DatePicker uses Value. The automatic variable always follows the current control ID. Use one owner for updates and consult the function reference for the method index conventions.

Troubleshooting and known boundaries

Symptom	Check / response
No procedure runs	Confirm the actual event binding, exported local routine, slots module and three-parameter signature.
Undefined renamed control variable	Inspect the accepted .kform and managed declarations together; deliberately update custom references and run project check.
Duplicate ID	Compare IDs case-insensitively across all forms sharing the slots module. Do not create a second matching PUBLIC symbol.
MAP has no callable member	Use only supported native methods. Browser JavaScript convenience methods are not automatically Kokum APIs.
NUMBER cannot assign to INTEGER	Use INT(value) after NULL/whole-number validation; do not change every ID declaration to NUMBER.
DatePicker appears but GetDate fails	Use form.<id>.GetDate(mask), a GUI1.9-capable VM and a newly compiled bundle; the generated value is a STRING.
New widget missing after update	Restart the IDE, update its served assets and hard-refresh. Existing KAPP resources do not update when only the VM changes.
Tab content missing	Verify that content is parented to the intended TabPage.
Slow TableView refresh	Assign a result array once or use a consistent method-owned path; avoid alternating ownership and unnecessary per-cell full updates.
Timer or GUI seems blocked	Avoid blocking Connect/long SQL or Result waits in owner slots; poll completed tasks/queues with bounds.

Open shared GUI findings

Known issue	What not to assume
TableView array handoff	Two displayed ARRAY rows can still yield native RowCount=0 and SetCell=false in a later slot. A method-only example passing does not fix the handoff.
Splitter.Position	A native Position patch may not move the rendered divider, despite working user dragging.
TextBox.Visible	The tested hidden flag can be overridden by an existing display style. DatePicker-specific visibility handling does not repair TextBox.
Dynamic Image	Correct native resource bytes/URLs do not guarantee stable rendered SetImage/Clear replacement.

The book deliberately retains these findings. They were not repaired during documentation authoring. Do not implement an unreviewed JavaScript override and then describe it as current Kokum behavior. Native Windows/macOS/provider acceptance is separate from Linux source and slot examples.

Safe deployment checklist

Save and check the project. Bundle with matching renderer resources and the required VM. Keep configuration and writable application data outside the bundle. Copy the complete Linux runtime directory, not only its .elf binaries. Test actual input, selection, errors, close/reopen and a relocated working path. Preserve application databases and user accounts when upgrading. [S15]

Sources, scope and verification

This edition describes the completed Kokum Linux source package below. It is application-developer documentation, not a C embedding API reference or a promise that every historical platform and renderer feature is complete.

Source identity

```
Kokum_Linux_1_1_0_MakeDist_Existing_Binaries.zip
SHA-256
9123f697fc536700a3d86926c8dc9548c5ddfa47b94e4bd01fa4d2d578f9561f
```

Product 1.1.0; VM ABI 1.7; browser GUI contract 1.9; bytecode/bundle format 1.3. Windows and macOS remain separate projects and have not adopted this Linux DatePicker/GetDate revision. No source code was edited for these books.

S01 — Built-in catalogue and language behavior

src/tlang_builtin_catalog.c; src/tlang_language_service.c; src/tlang_interpreter.c; src/tlang_ir_executor.c; src/tlang_vm_engine.c

S02 — Widget schema, control methods and form bridge

src/tlang_gui_schema.c; src/kokum_form_control_internal.h; src/tlang_web_form.c; src/tlang_form_variables.c; include/tlang_gui_schema.h

S03 — Designer workflow, events and automatic variables

MD/docs/BROWSER_VISUAL_FORM_DESIGNER.md; MD/docs/SLOT_BINDINGS.md; MD/docs/AUTOMATIC_WIDGET_VARIABLES.md; web/ide/kokum-form-designer.js

S04 — Scalar mathematics and array statistics

MD/docs/SCALAR_MATHEMATICS.md; MD/docs/ARRAY_STATISTICS.md; src/kokum_math.c; src/kokum_statistics.c

S05 — Files, regex and natural patterns

MD/docs/FILE_IO.md; MD/docs/PHASE4_16_7_NATURAL_PATTERN_OPERATIONS.md; src/kokum_fileio.c; src/kokum_regex.c; src/kokum_find.c

S06 — Tasks, named threads and synchronization

MD/docs/ASYNC_TASKS.md; MD/docs/PHASE4_14_2_NAMED_THREADS.md; MD/docs/PHASE4_14_4_SAFE_SHARED_STATE.md; src/kokum_task.c; src/kokum_sync.c

S07 — Database components and Remote FlatDB

MD/docs/DATABASE_COMPONENTS.md; src/kokum_database.c; src/kokum_flatdb.c; tests/run_kokum_029_flatdb_tests.py

S08 — Graph resource

MD/docs/GRAPH_WIDGET.md; src/kokum_graph.c

S09 — Image and WebSocket resources

MD/docs/IMAGE_WIDGET.md; MD/docs/WEBSOCKET_CLIENT.md; src/kokum_image.c; src/kokum_websocket.c

S10 — REST and HTTP

MD/docs/REST_REQUEST_EXECUTION.md; MD/docs/REST_SECURITY_FAILURES_CONCURRENCY.md; MD/docs/URL_ASYNC_MICROSERVICES_IDE.md; src/kokum_url.c

S11 — TOTP

MD/docs/TOTP_BUILTIN.md; src/kokum_totp.c; src/kokum_totp_crypto.c

S12 — String trimming

src/kokum_string_trim_internal.h; tests/features/string_trim/; Kokum_String_Trim_Guide.md supplied with the trimming release

S13 — DatePicker and GetDate

Current DatePicker/GetDate guides, feature tests and src/kokum_form_control_internal.h; src/tlang_gui_schema.c

S14 — Local and network IDE administration

MD/docs/LAN_CSV_AUTHENTICATION.md; MD/docs/LAN_PROJECT_OPERATIONS.md; MD/docs/LAN_REMOTE_WORKSPACES.md; config/kokum-ide-lan.conf.example

S15 — Current binary distribution

Root make_dist; Kokum_MakeDist_Existing_Binaries_Guide.md, distributed with the current source edition

S16 — Backend services and project operations

src/kokum_backend.c; examples/kokum_029_backend_services/; current kokumc and kokumvm command help

S17 — Reference example foundations

Kokum_Complete_Command_Reference_0_29_0.docx; earlier GUI Designer manual screenshots. Entries were checked against the current source and newly tested where stated.

What was checked for this edition

The unchanged current Linux compiler, VM, IDE and FlatDB server were built out of tree. All 244 callable examples in the companion function reference passed compiler checking. Of these, 241 were executed: 133 through the source, IR, bytecode and external-VM paths; 103 through compiled GUI slots; and five against a real local Kokum FlatDB server. Three server-specific database examples were compiler-checked only and require an operator-supplied remote test database.

GUI-slot execution verifies native calls and protocol results; it is not a claim that every widget was visually tested in this documentation pass. Earlier acceptance evidence, static source contracts and current executable examples are distinguished in the text. Synthetic/local networking is not production broker or database-server acceptance. A check can pass while a separately documented renderer limitation remains open.

The documents were generated separately from the source checkout. DOCX pagination and navigation were rendered and inspected. No native Windows/macOS run, full product regression, clean-machine distribution test, comprehensive browser/accessibility test or new security certification is asserted.

A-Z widget and topic index

Page numbers refer to this edition. Entries are linked; use Ctrl+F for an exact function, control or method name.

Entry	Section	Page
About this edition	Guide	3
Asynchronous GUI work	Lab	17
Button	Widget	21
CheckBox	Widget	22
ComboBox	Widget	23
Common properties and event directory	Guide	59
Create your first Designer application	Guide	7
Database table population	Lab	16
DatePicker	Widget	24
DatePicker.GetDate	Lab	15
DockPanel	Widget	25
Graph	Widget	26
Image	Widget	27
Install and start the Linux tools	Guide	4
Label	Widget	28
Layouts, themes and multiple forms	Guide	13
ListBox	Widget	29
ListView	Widget	30
Menu	Widget	31
MenuBar	Widget	32
MenuItem	Widget	33
ModalDialog	Widget	34
Network IDE and user administration	Guide	5
Panel	Widget	35
PopupMenu	Widget	36
PostgreSQL	Widget	37
Practical labs	Guide	15
ProgressBar	Widget	38
PropertyGrid	Widget	39
PUBLIC variable ownership	Concept	11
RadioButton	Widget	40
Renaming a control	Workflow	9
ScrollView	Widget	41
Signals, slots and generated state	Guide	11
Sources, scope and verification	Guide	61
Splitter	Widget	42
SQLite	Widget	43
StatusBar	Widget	44
SubForm	Widget	45
SubFormHost	Widget	46
TableView	Widget	47
TabPage	Widget	48

Entry	Section	Page
TabView	Widget	49
TextArea	Widget	50
TextBox	Widget	51
TigerDB	Widget	52
Timer	Widget	53
ToolBar	Widget	54
ToolButton	Widget	55
TreeView	Widget	56
Troubleshooting and known boundaries	Guide	60
WebSocket polling	Lab	18
WebSocketClient	Widget	57
Widget reference A-Z	Guide	19
Window	Widget	58
Work efficiently in the Designer	Guide	9