Re_forms 21

RIB — RE_FORMS21 INFORMATION BASE

Stop Flying Blind in Your Re_Forms21 Codebase

Your system shouldn't depend on who's still around to remember how it works.

— THE CHALLENGE

Enterprise teams running Re_Forms21 systems are sitting on years of undocumented complexity. UI forms are wired to deep PL/SQL logic that lives only in senior developers' heads. One schema change can silently break a dozen downstream procedures — and nobody finds out until production does.

The result: refactoring is avoided, onboarding takes months, and every modernization effort stalls because the system can't be trusted.

— HOW RIB SOLVES IT

RIB ingests your Java source, PL/SQL, Oracle schemas, and UI metadata — then builds a precise, queryable map of every dependency, data flow, and execution path across your entire Re_Forms21 application.

It combines deterministic static analysis with semantic embeddings so developers, QA teams, and AI agents always work from authoritative system context — not guesswork.

— WHY RIB

The map that replaces memory
Cross-layer dependency graphs replace what used to live in senior devs' heads.

Safe refactoring, every time
Know exactly what breaks before you touch a table, class, or procedure.

New devs productive in days, not months
On-demand architectural insight, not a six-month archaeology project.

AI that actually knows your system
Feeds Cursor, Claude Code, and other copilots with real structural context via MCP, CLI, and Cypher QL.

— WHAT YOU CAN DO WITH IT

WORKFLOW	WHAT IT MEANS
Dependency Tracing	Find every Java, SQL, PL/SQL, and UI dependency of any object or method — instantly
UI-to-Database Backtracking	Trace any screen back to the exact variable and procedure that drives it
Impact Analysis	Before touching anything, see the full blast radius across layers
Copilot Empowerment	Feed authoritative context directly into your IDE — no hallucinated answers

— THE BUSINESS CASE

	WITHOUT RIB	WITH RIB
Understanding a dependency chain	Days — sometimes weeks	**Minutes**
Risk of regression from refactoring	High — blast radius unknown until production	**Low — full map before you touch anything**
New developer ramp time	3–6 months	**Days to first contribution**
AI agent accuracy on your codebase	Low — generic context, frequent hallucinations	**High — deterministic system graph**

— WHO IT'S FOR

- Engineering leads modernizing or maintaining Re_Forms21 systems
- Architects who need cross-layer visibility before committing to a refactoring plan
- QA teams building test automation against complex PL/SQL and Java dependencies
- Any team using AI copilots that keep hallucinating your architecture

GET STARTED

RIB connects to your Git repositories and Oracle Database in a single session — no codebase changes, no infrastructure setup.

→ Request a technical walkthrough — solutions@reforms21.com

→ See it in your environment — reforms21.com/rib-demo

RIB is part of the **ReML** suite — the AI-powered modernization platform for Re_Forms21 applications.