Consultingwerk
software architecture and development

Cool Things Your Coding Assistant Can Do For You

Mike Fechner
Consultingwerk

The full stack modernization framework
SmartComponent Library
by Consultingwerk

Mike Fechner



- 36 years of experience in Progress (from Version 5 to OpenEdge 13)
- Active member of the OpenEdge community and speaker at international conferences
- Expert in software architecture, object-oriented design, and web technologies

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Consultingwerk
software architecture and development

Our Focus: Modernizing OpenEdge Applications



Modernization of Legacy OpenEdge Applications



Deep Technical Expertise



Global IT Partner with Local Presence



More than Consulting – We Deliver Tools & Solutions

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Working with Jira Tickets

From the meeting to the ticket — and from the ticket to the code

Live Demo 1: From Meeting Notes to Jira Tickets



Before

- Customer meeting last week
- Recorded by Read AI, our AI meeting notetaker
- Ideas buried in the transcript

```
» Prompt:
using read.ai MCP server, I had a
meeting with John from <customer>
last week. Did we discuss any
potential improvements to the
SmartComponent Library/Framework?

» Follow-up:
make sure the tickets have the
<customer> label and due date end
this week.
```

You'll see

- Claude asks Read AI via MCP
- Improvement ideas from the meeting
- Jira tickets with label and due date

Without the agent: re-read the notes, then type every ticket by hand

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Live Demo 2: Analyzing a Real-World Ticket

Before

- Story for our Angular client: a per-node “checkable” flag
- The desktop variant needs it too: GUI for .NET with the Infragistics UltraTree

```
» Prompt:
<ticket> - the per node checkable flag
Radu asks for - can we apply that to
the UltraTree nodes too? As we need
that for the Desktop variant too? Then
check for the overrides properties of
the UltraTreeNode. Do not create a
ticket yet - just analyze that for me.

» Follow-up:
Radu also proposed per node visibility
as an option. How about that?
```

You’ll see

- Claude reads the story and the comments
- Maps it to the UltraTreeNode API
- An analysis — no ticket, no code

Analysis first: the agent does the homework — I make the decision

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Fixing Sonar Issues

Static analysis finds them — the agent fixes them

Status
Security Category
Creation Date
Language

Rule 1

Search for rules...

(OpenEdge) Unused variables and ... 15
(OpenEdge) FIND FIRST vs. unique i... 3
(OpenEdge) Block label 2
(OpenEdge) Empty code block 2
(OpenEdge) Enforce uppercase or l... 2
(OpenEdge) Redundant CATCH stat... 2
(OpenEdge) FIND and GET EXCLUSI... 1
(OpenEdge) Whole index 1

Select issues Navigate to issue **3** issues **15min** effort

Consultingwerk/.../Repository/Object/ObjectMasterDataAccess.cls

FIND expression doesn't need 'FIRST' qualifier as the WHERE clause will return a unique result

Code Smell Low rssw xref

Open Mike Fechner L645 • 5min effort • 10 days ago

Consultingwerk/SmartFramework/Repository/SmartRepositoryIntegrityCheckTask.cls

FIND expression doesn't need 'FIRST' qualifier as the WHERE clause will return a unique result

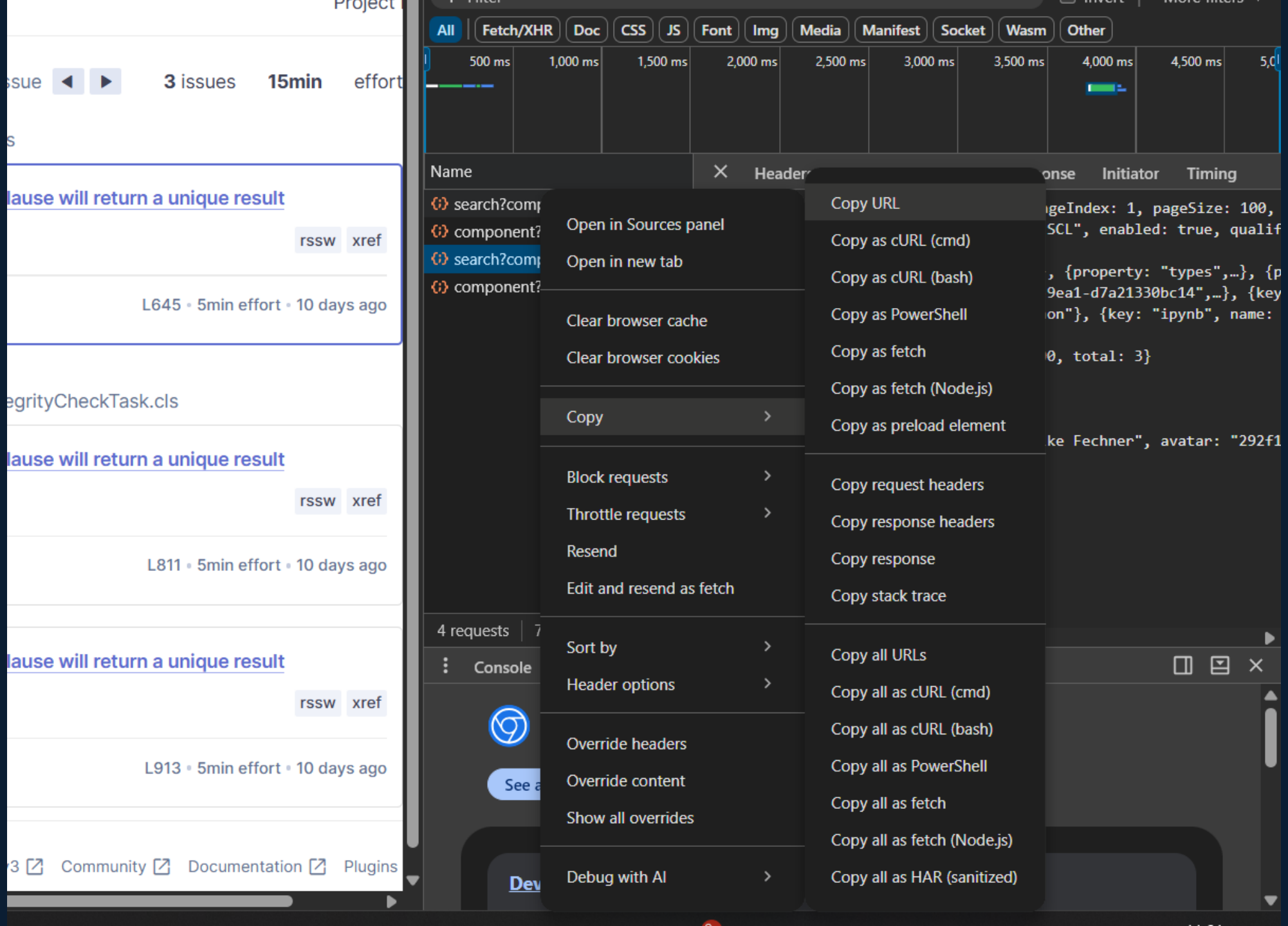
Code Smell Low rssw xref

Open Mike Fechner L811 • 5min effort • 10 days ago

FIND expression doesn't need 'FIRST' qualifier as the WHERE clause will return a unique result

Code Smell Low rssw xref

Open Mike Fechner L913 • 5min effort • 10 days ago


3 issues
15min
effort
lause will return a unique result
rssw
xref
L645 • 5min effort • 10 days ago
egrityCheckTask.cls
lause will return a unique result
rssw
xref
L811 • 5min effort • 10 days ago
lause will return a unique result
rssw
xref
L913 • 5min effort • 10 days ago
Community
Documentation
Plugins
All
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Manifest
Socket
Wasm
Other
500 ms
1,000 ms
1,500 ms
2,000 ms
2,500 ms
3,000 ms
3,500 ms
4,000 ms
4,500 ms
Name
Initiator
Timing
search?comp
component?
search?comp
component?
Open in Sources panel
Open in new tab
Clear browser cache
Clear browser cookies
Copy
Block requests
Throttle requests
Resend
Edit and resend as fetch
Sort by
Header options
Override headers
Override content
Show all overrides
Debug with AI
Copy URL
Copy as cURL (cmd)
Copy as cURL (bash)
Copy as PowerShell
Copy as fetch
Copy as fetch (Node.js)
Copy as preload element
Copy request headers
Copy response headers
Copy response
Copy stack trace
Copy all URLs
Copy all as cURL (cmd)
Copy all as cURL (bash)
Copy all as PowerShell
Copy all as fetch
Copy all as fetch (Node.js)
Copy all as HAR (sanitized)
4 requests
Console

Live Demo 3: Fix the Sonar Issues

```
» Prompt:
Fix linting issues from SonarQube:
http://<sonarqube>:9000/api/issues/search?components=Consultingwerk%3ASCL&s=FILE_LINE
&rules=rssw-oe-main%3Aeu.rssw.antlr.proparse.checks.FindFirst&inNewCodePeriod=true
&issueStatuses=CONFIRMED%2COPEN&ps=100&facets=impactSoftwareQualities%2Cseverities
%2Ctypes%2CimpactSeverities%2Crules&additionalFields=_all&timeZone=Europe%2FBerlin

Commit straight into develop, no ticket needed
```

Before

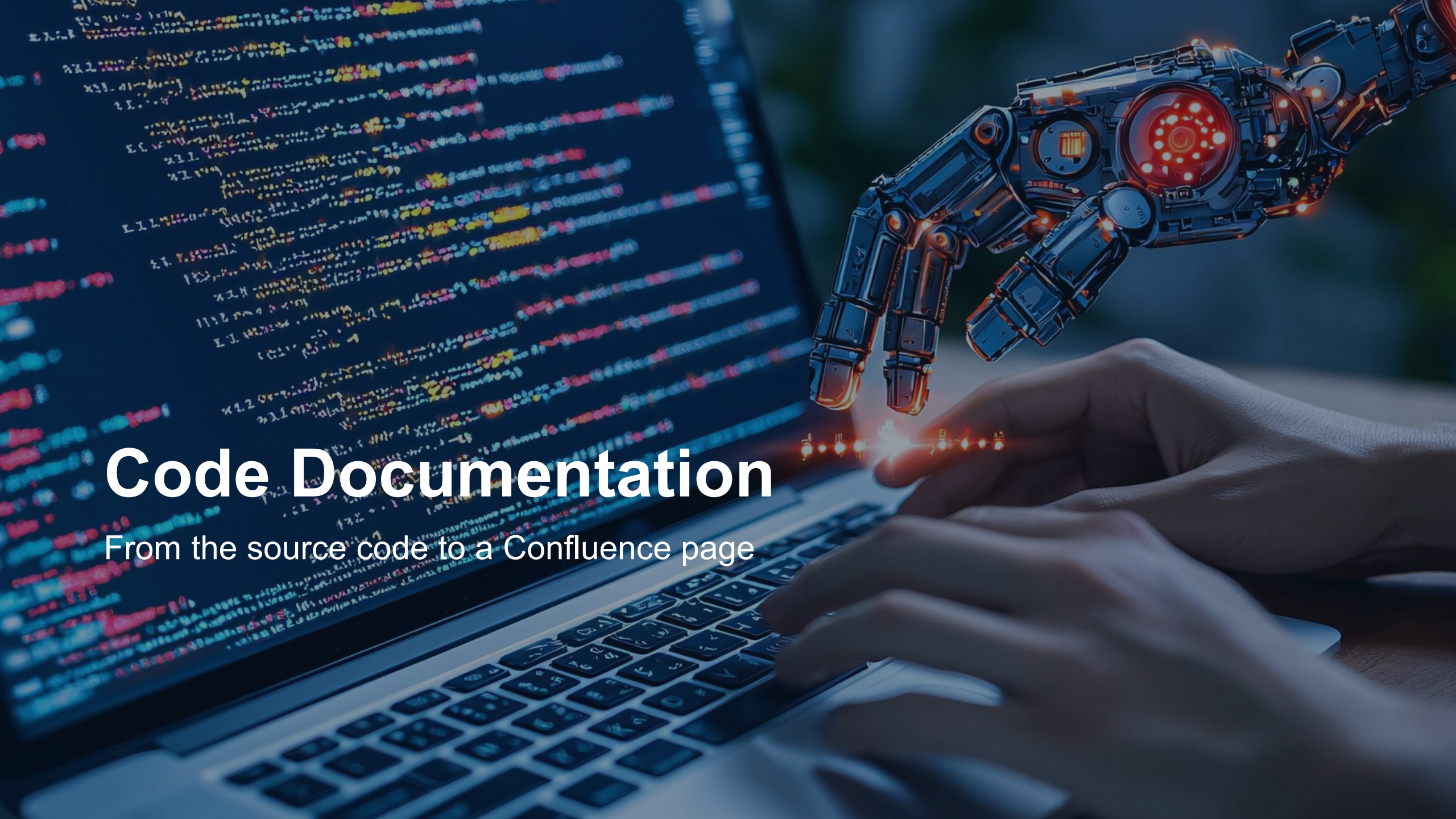
3 × “FIND FIRST vs. unique index” in new code — 5 minutes each, by hand

You’ll see

Claude fixes the code, runs the relevant unit tests to catch regressions, compiles and commits to develop

SonarQube MCP server in Docker? Often overkill — DevTools → Copy URL is enough

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Code Documentation

From the source code to a Confluence page

Live Demo 4: Document a Business Entity

```
» Prompt:
In my Personal Confluence space, please document the
Consultingwerk.SmartComponentsDemo.OERA.Sports2000.OrderBusinessEntity
* include the ERD diagram (image PNG) of the Business Entity Designer
* include description of schema, tables with relations
* include overview of exposed RESTful endpoints
* include description of custom methods
* include description of validation
```

Before

A Business Entity — the docs are somewhere, and outdated

You'll see

A Confluence page: ERD, schema, REST, methods, validation

Generation takes a few minutes — here is one I prepared earlier: **open the Confluence page**

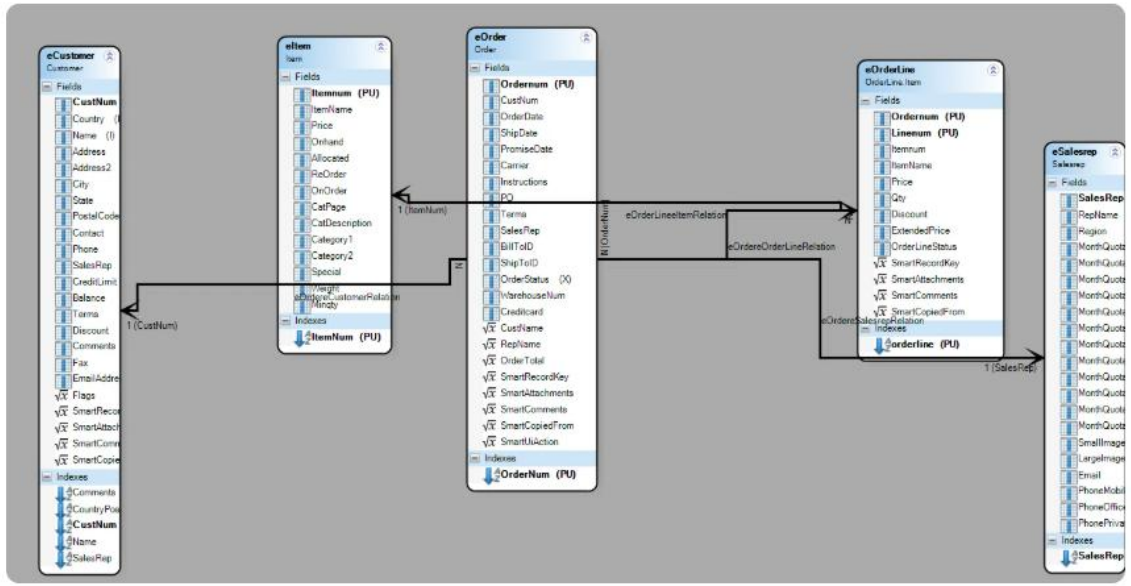
© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

https://consultingwerk.atlassian.net/wiki/spaces/~5570586e3bc2a2bee640f29d9c3df610f16681/pages/4151443458/OrderBusin...
Neue Chrome-Version verfügbar
SharePoint-Dokumente und -Dateien durchsuchen
Erstellen
Upgrade durchführen
Rovo fragen
9+
OrderBusinessEntity (SmartComponentsDemo Sports2000)
Aktualisiert: jetzt
Bearbeiten
Teilen

Aspect	Value

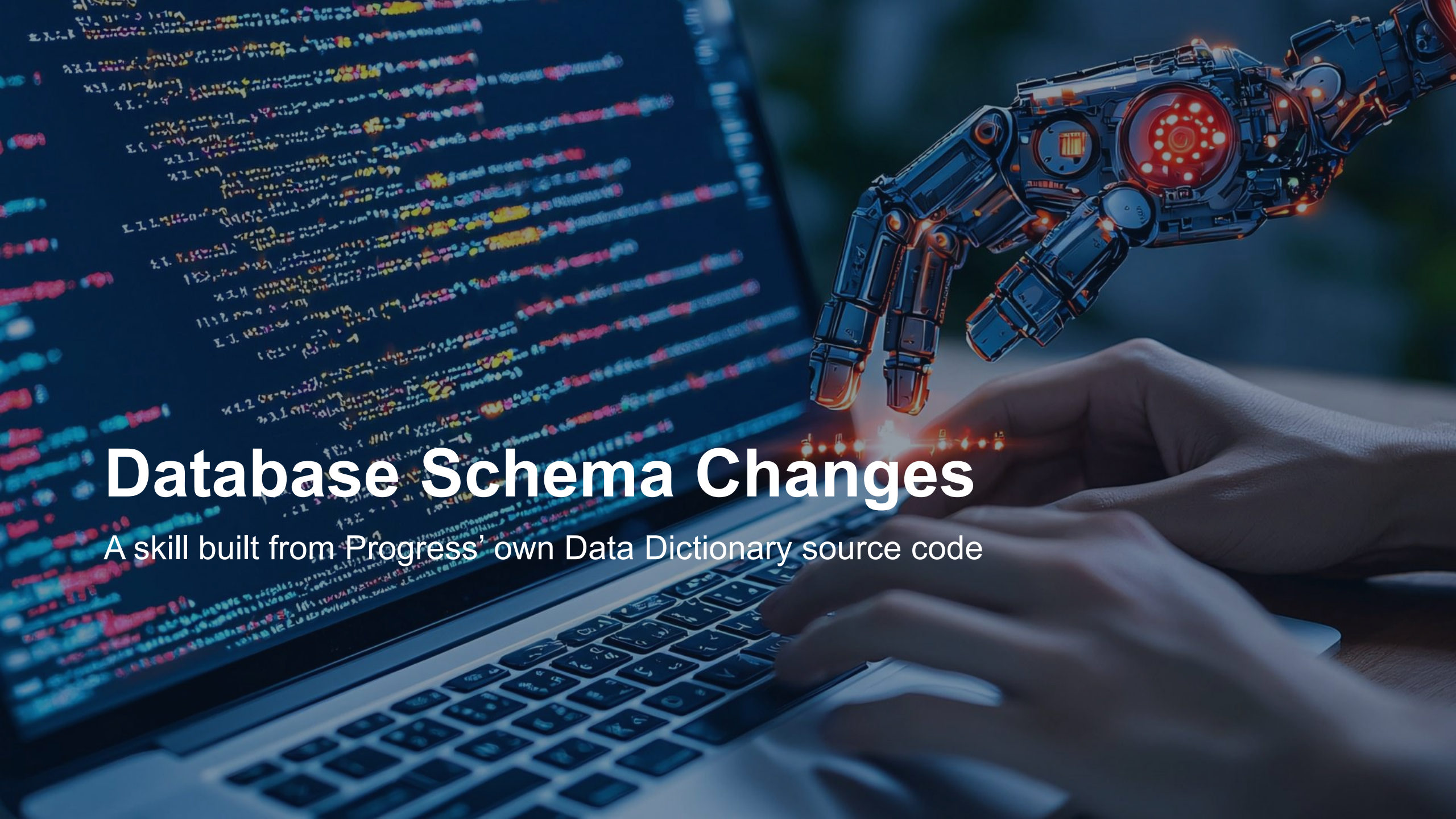
Entity Relationship Diagram

The diagram below is taken from the Business Entity Designer (`OrderBusinessEntity.png`). Each box is a temp-table of the ProDataset with its source database table(s) underneath the name; fields marked with a root sign are calculated fields without a database source, (PU) marks the fields of the primary unique index.



Schema

Temp-tables



Database Schema Changes

A skill built from Progress' own Data Dictionary source code

Teaching the Agent the .df Format



The source: Progress' Data Dictionary (ADE source code)

`prodict/dump/_dmpdefs.p` writes a .df · `_lodsddl.p` parses and loads it · `_dmpincr.p` builds a delta

Refactored by the agent into a skill reference: format grammar, authoring guide, incremental .df

The workflow the skill enforces



The documentation is the source code — the agent read it, so I don't have to

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Two Skills: Generic and SmartDB



update-database-schema

- Any database, by its short name
- Storage areas from the `.st` file
- Data types from the field name suffix
- Drops only after an explicit yes
- Target database confirmed before every load



update-smartdb-database-schema

- Builds on the generic skill — and wins where they differ
- Always SmartDB · table prefix `Smart`
- GUID field first + primary unique index
- Unspecified fields copied from the existing `SmartDB.df`
- `-SA "U"` string attributes · re-dump the full schema

Generic rules once — project conventions as a thin layer on top

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Live Demo 5: A New SmartDB Table



```
» Prompt:
Create a new branch feature/Demo-Schema and a SmartDB for that branch.

Lets add a new database table in the SmartDB Database for an universal word index (HyperIndex).

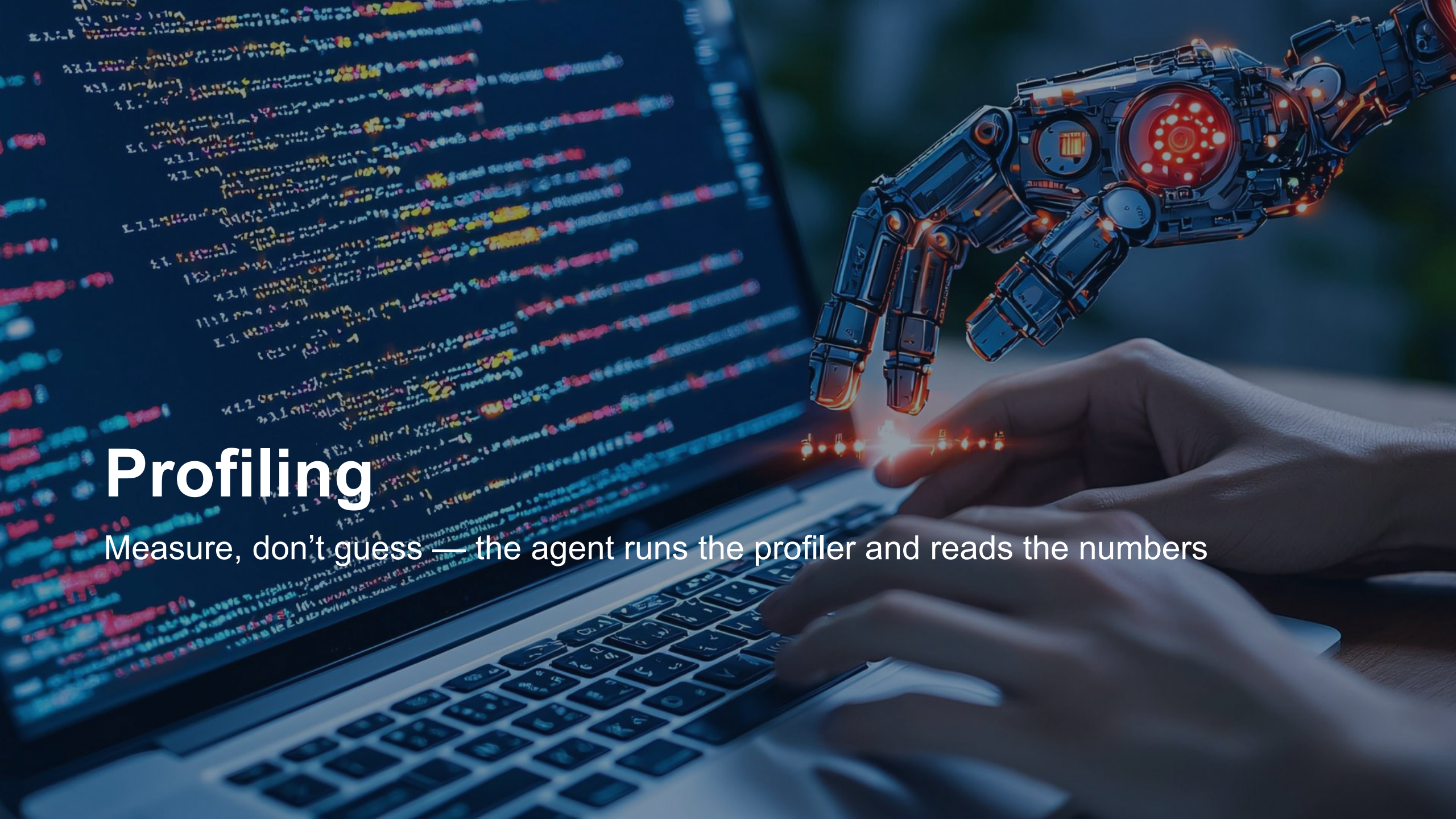
Table name: SmartHyperIndex

Fields:
* TableGuid (reference to SmartTable)
* KeyField values (key fields for a record in the application DB as defined in the SmartTable record)
* IndexValue (CHARACTER), a word-indexed field

FK index on TableGuid
```

You'll see: a delta .df with GUID field and primary unique index — then the agent asks before it loads

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Profiling

Measure, don’t guess — the agent runs the profiler and reads the numbers

The Skill: unit-test-profiler-harness

Temporary harness test class → Warm-up run → Scoped profiler session → Own output folder per run → Analyzer: rank and diff



Stop the runner's profiler

The PCT test runner starts the AVM with `-profile` — the harness switches it off first



Before and after

Every run gets its own folder (`-1`, `-2`, …) — the analyzer diffs them



Ratios, not milliseconds

`-profile` slows the session — wall-clock timings in a bare session with `etime`

Temporary tooling — never committed, results outside the workspace

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Live Demo 6: Measure a Refactoring



```
» Prompt:
Create a simple procedure Demo\ProfilingSample\customer-orders.p
It receives a salesrep as input and contains a FOR EACH Customer where Salesrep .... and a nested
FOR EACH for orders of that customer as nested (separate) FOR EACH joining orderlines of the order
and calculates the total of OrderLine Qty * Price for every Order.
The total order value is returned as an output parameter.
In the procedure, limit to 5000 customers of the salesrep

» Follow-up:
I want to refactor this now to use an internal procedure for the inner FOR EACH - receiving a
CustNum as input and returning the total order value for a customer as an output parameter.
So write a second procedure customer-ordersv2.p and use the profiler harness to compare the
runtime of both
```

You'll see: two procedures, a harness, a profiler run — and a verdict backed by numbers

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Result: the Refactoring Costs 70 %

Result: v2 (with the internal procedure) is about 70% slower than v1. Both return the same total, 2,370,534,070.7.

Each variant was warmed up once, then run 10 times for BBB. Every run reached the 5000-customer limit, so there were 50,000 calls of `get-customer-order-value` in total.

	v1 `customer-orders.p`	v2 `customer-ordersv2.p`
Wall clock (`etime`)	2,085 ms	3,540 ms
Profiler total	2.01 s	3.45 s
Order/OrderLine `for each` (705,650 iterations)	0.64 s	0.88 s
`assign … Qty * Price` (655,650 executions)	0.25 s	0.32 s
The `run get-customer-order-value` statement itself (50,000 calls)	-	0.45 s

Where the extra time comes from:

- **The call itself:** about 9 µs per `run` under the profiler, 0.45 s in total.
- **Opening the query on every call:** the same Order/OrderLine loop costs about 35% more in v2. In v1 the buffers stay open for the whole procedure; in v2 the order query is set up again for every customer, and the loop's own statements carry that cost.



Decided on data, not on gut feeling

- Same total in both versions
- The `run` itself: ~9 µs per call
- The real cost: the order query is opened again for every customer
- **Clean code is not free — measure it**

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Why Is the Build Red?

Analyzing a failing Jenkins build

Live Demo 7: Analyze a Failing Build



```
» Prompt:
Why is this failing on 13.0 - it's OK on 12.8:
https://<jenkins>/job/13.0/job/Git/job/develop/lastCompletedBuild/
```



Before

develop on OpenEdge 13.0 is unstable, 12.8 is green — one of 29,000+ tests is red





You'll see

Claude reads build, test report and history — test, cause and fix, then asks

From a red build to the root cause — in one prompt

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

The Answer

Why 13.0 `develop` #1531 is failing

The build is **UNSTABLE**, with 1 failed test out of 29,112. The same test has failed in every completed build since #1527 (#1522–#1526 don't appear in the build list); #1521 was the last green one.

- **Failing test:** `Consultingwerk.BusinessEntityGeneratorTest.SCL[redacted].TestGeneration:TestQueryGeneration`.
- **Error:** it expects the `PlanningType` property of the generated `SmartSchedulerJobPlanQuery` to have the type `…SmartSchedulerJobPlanQueryPlanningTypePredicate`. It actually gets `GenericCharacterEnumPredicate<SmartSchedulerJobPlanQuery, SchedulerJobPlanningTypeEnum>`.

Cause:

- **The generator is right.** `AbstractTableQueryBuilderGeneratorPlugin` writes the predicate types behind `&IF DEFINED (OpenEdge_13_0)`. On 13.0 and later, the generated query uses the generic predicate types. Before 13.0, it uses the dedicated predicate classes.
- **The test isn't version-aware.** It only knows the pre-13.0 type names.
- **Why it only shows up now:** these tests never ran on Jenkins until your SCL-[redacted] commits `4eb3b688d` and `179a53753` added the "Business Entity Generator Ticket Tests" suite to the build. On 12.8 the same test passes.

Fix: the three `Assert:Equals (…, oProperty:DataTypeName)` checks in TestGeneration.cls:88 need an `&if defined (OpenEdge_13_0)` branch that expects the generic type names.

I haven't changed anything. Should I make the fix on `feature/SCL-[redacted]`, or as a new Bug if that ticket is already closed?

What the agent did

- Read the build, the test report and the build history
- Found the `&IF DEFINED (OpenEdge_13_0)` in the generator
- Traced it to the commits that added the test suite
- Changed nothing — and asked where the fix belongs
- **The generator is right — the test isn't version-aware**

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



From JSON Schema to ABL

A schema in — typed classes with validation out

Introduction

Get Started

Overview
What is a schema?
The basics
Create your first schema
Tour of JSON Schema
JSON Schema glossary

Examples

Miscellaneous examples
Modelling a file system
Other examples

Guides

Reference

Specification

Calendar ¶

A schema representing an event in a calendar, including properties like `startDate`, `endDate`, `summary`, `location`, and `recurrenceDate` details. The `geo` property is a reference (`$ref`) to another schema defined at a different location which represents a geographical location with latitude and longitude values.

schema

```
{
  "$id": "https://example.com/calendar.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "description": "A representation of an event",
  "type": "object",
  "required": [ "dtstart", "summary" ],
  "properties": {
    "startDate": {
      "type": "string",
      "description": "Event starting time"
    },
    "endDate": {
      "type": "string",
      "description": "Event ending time"
    },
    "summary": {
      "type": "string"
    },
```

Open page

Live Demo 8: JSON Schema to an ABL POJO



Before

- A JSON Schema from the spec website
- Calendar event + geo location
- Usually: classes, parsing and checks by hand

```
» Prompt:
based on the JSON Schema below, create
an ABL POJO in Demo.JsonSchema with a
Reader and Writer class supporting the
validation from the JSON schema -
include the geo location

<paste schema here>
```

You'll see

- Typed classes for event and geo location
- Reader and writer with the schema's rules
- Required fields, min / max checks

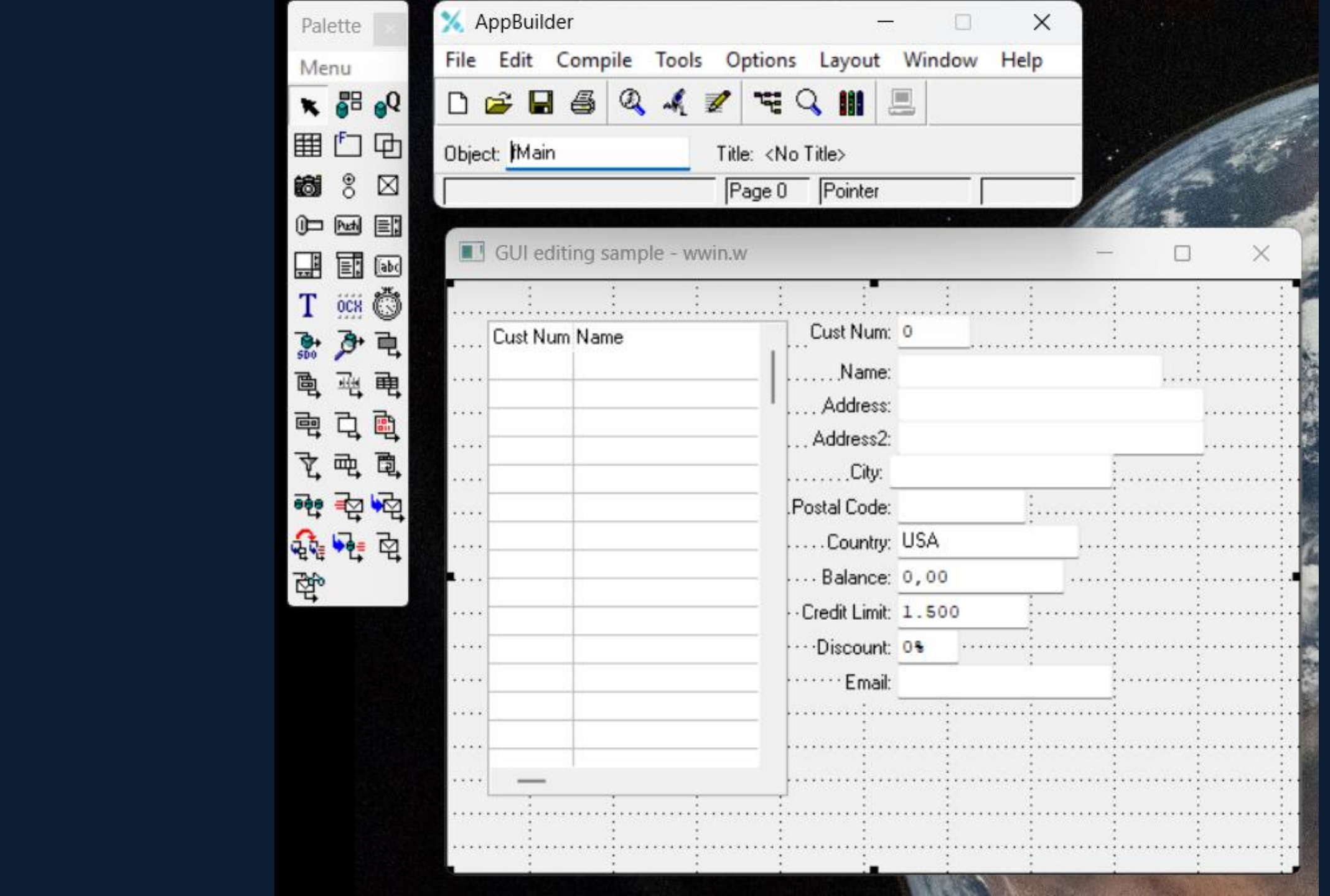
The schema is the spec — the agent turns it into code

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Say Bye-Bye to the AppBuilder

Editing legacy GUI windows — and keeping the AppBuilder markup intact


Palette
Menu
AppBuilder
File Edit Compile Tools Options Layout Window Help
Object: Main
Title: <No Title>
Page 0
Pointer
GUI editing sample - wwin.w
Cust Num Name
Cust Num: 0
Name:
Address:
Address2:
City:
Postal Code:
Country: USA
Balance: 0,00
Credit Limit: 1.500
Discount: 0%
Email:

Live Demo 9: Layout Without the AppBuilder



```
» Step 1 — wider browse, new columns, even spacing:
Demo\GuiEditing\wwin.w is an ABL legacy GUI program - when editing maintain the AppBuilder markup.
Attached a screenshot - please make the browser twice as wide and add Salesrep and Country to it.
Move the fill-ins to the right accordingly. Make sure they all have an even distance to each other.

» Step 2 — rectangles and tab order:
put a rectangle around the address fields including name and a rectangle around the commercial terms.
Move email address up to below the customer number. Ensure tab-order is from top down.

» Step 2b:
add the labels "Address" and "Commercial terms" to the rectangles
```

You'll see: the window changes — and still opens in the AppBuilder

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Live Demo 9: Behavior Without the AppBuilder



```
» Step 3 — row coloring:
implement row-display trigger coloring the customers from USA with BGCOLOR 12

» Step 4 — resizing:
Implement resizing. min size should be the startup size, enable resizing of the
window. In the window-resized trigger, first ensure frame is large enough, resize
grid first (grow in height and width) and move the fill ins to the right if
required. When shrinking, first shrink the field level widths and the browse,
then the frame.
```

You'll see: triggers in the right AppBuilder sections — and a window that finally resizes

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

My Sessions at PUG Challenge Providence

Session	Day	Time	Room
Say Bye-Bye to the IDE. Here Comes Claude Code.	Mon 5	1:00 – 2:00 PM	Mozart
AI-Powered OpenEdge Modernization: Faster, Smarter, Scalable	Mon 5	3:45 – 4:45 PM	Beethoven
REST API Versioning: Concepts and Samples	Tue 6	8:30 – 9:30 AM	Mozart
Cool Things Your Coding Assistant Can Do For You	**Tue 6**	**1:15 – 2:15 PM**	**Mozart**

Highlighted: this session

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Subscribe Now to Consultingwerk's Official Newsletter!

www.consultingwerk.com/resources/newsletter

or marketing@consultingwerk.com

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Questions



© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.


Consultingwerk
software architecture and development