

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

LESSON PLANS

**SL 3**



---

**LESSON PLAN OUTLINES — SL 3 ADVANCED  
BUILDER**

---

*USAREUR-AF Operational Data Team — C2DAO*

HEADQUARTERS  
UNITED STATES ARMY EUROPE AND AFRICA  
(USAREUR-AF)  
Wiesbaden, Germany

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

**26 MARCH 2026**

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

# LESSON PLAN OUTLINES — SL 3 ADVANCED BUILDER

## USAREUR-AF OPERATIONAL DATA TEAM — C2DAO

**Course:** SL 3 Advanced Builder | **Duration:** 5 days (40 hours) | **Version:** 1.0 — March 2026

Abbreviated LP format. Expand using [LP\\_TEMPLATE.md](#) as needed. Instructor prerequisite: SL 4-level (any track) or C2DAO SME designation. T:I ratio 6:1.

## AUTHORITATIVE REFERENCES

Publication	Title	Relevance
AR 350-1	Army Training and Leader Development	Master regulation for Army training policy; governs lesson plan standards and instructional requirements
FM 7-0	Training	Unit training management procedures; context for how SL 3 advanced builder training integrates with unit training plans
AR 25-1	Army Information Technology	Establishes data governance authority; relevant to SL 3 governance and data steward review content

## DAY 1 — ADVANCED WORKSHOP

### Block 1 — Multi-Page Workshop

**Hours:** 2.0 | **Method:** Lab | **Time:** 0830–1030

**Purpose:** SL 3 Workshop applications serve multiple user types from a single application — commanders see portfolios, analysts see details, logisticians see specific views. Multi-page applications require explicit design for navigation and data flow.

**TLO:** Given a scenario, the trainee will build a multi-page Workshop application with page navigation controls, page-level parameters, and at least two URL deep links — to standard.

**Key Delivery Notes:** - Page navigation: add pages from the Workshop page management panel. Each page has its own widget layout. Navigation links between pages can be button widgets or hyperlink widgets. - Page parameters: each page can accept parameters that filter or scope its data. Parameters are passed from a source page via navigation. Configure page parameters in the page settings before building widgets. - URL deep linking: Foundry Workshop URLs encode current page and parameter state. Trainees should test that their deep link navigates directly to the correct page with the correct parameter active.

**Student Activity:** Build a 3-page application skeleton (no data yet): Portfolio page → Unit Detail page → Historical Trend page. Configure navigation links between pages. Test that navigation works.

**Assessment:** Evaluated as component of Practical Exercise Task 4 (multi-page application).

---

## Block 2 — Conditional Logic

---

**Hours:** 1.25 | **Method:** Lab | **Time:** 1045–1200

**Purpose:** Conditional logic makes Workshop applications context-aware — showing status banners when data is bad, hiding sections that aren't relevant, color-coding tables by status.

**TLO:** The trainee will configure show/hide visibility rules on panels, apply conditional formatting to a table, and set up dynamic widget visibility based on an Object property value.

**Key Delivery Notes:** - Show/hide panels: configure visibility rule on a panel widget — show only when a variable or Object property meets a condition. Use case: "show the C4 warning banner only when at least one unit has C4 equipment." - Conditional table formatting: row color coding based on `c_rating` or `status`. RED for C3/C4, AMBER for C2, GREEN for C1. Configured in the table widget format settings. - Dynamic widget visibility: a chart may be relevant only on certain pages or when certain filters are active. Configure using visibility conditions.

**Student Activity:** Add conditional formatting to a table widget (color-code rows by status). Add a warning banner panel visible only when the Object count for a status condition exceeds a threshold.

**Assessment:** Evaluated as component of Practical Exercise Task 4.

---

## Block 3 — Variable Passing Between Pages

---

**Hours:** 2.0 | **Method:** Lab | **Time:** 1300–1500

**Purpose:** A Portfolio page that links to a Unit Detail page must pass the selected unit as context to the detail page. Without variable passing, the detail page has no way to know which unit the user clicked.

**TLO:** The trainee will configure variable passing between two pages — Object selection on Page 1 drives a filtered view on Page 2 — and verify that changing the Page 1 selection updates Page 2 content.

**Key Delivery Notes:** - Object selection as a page variable: when a user clicks a row in a Page 1 table, the selected Object becomes a page variable. - Pass the variable to Page 2 as a navigation parameter: navigate to Page 2 with `selected_unit_id = {Page1.selected_object.unit_id}`. - Page 2 receives the parameter and uses it to filter its table widget: `Filter: unit_id = {page_parameter.selected_unit_id}`. - Test: select Unit A on Page 1, navigate to Page 2, confirm only Unit A's records appear. Then go back, select Unit B, navigate to Page 2, confirm the view changed.

**Student Activity:** Connect Page 1 portfolio table to Page 2 detail view using variable passing. Test with at least 3 different unit selections.

**Common Errors:** - Page 2 not receiving the parameter (configuration step missed) — Page 2 shows all data regardless of selection; verify parameter passing in navigation configuration - Parameter name mismatch between sender (Page 1) and receiver (Page 2) — data doesn't filter; confirm names match exactly

**Assessment:** Evaluated in Practical Exercise Task 4 (multi-page navigation).

---

## Block 4 — Design Exercise: 3-Page Operations Dashboard

---

**Hours:** 1.75 | **Method:** Workshop | **Time:** 1515–1700

**Purpose:** Applying Blocks 1–3 to a defined scenario under instructor critique. This is the first instructor-observed design exercise — it reveals design thinking gaps before the complex Days 3–5 work.

**TLO:** Given a scenario, the trainee will build a 3-page dashboard (portfolio → unit detail → historical trend) meeting the design exercise requirements — reviewed by instructor against SL 3 design standards.

**Key Delivery Notes:** - Provide the scenario brief at the start of the block. Trainees have 10 minutes to plan on paper before opening the tool. - Instructor circulates and provides real-time critique: "Why is your filter widget here and not above the table?" "What does this metric show when zero units have C4?" - Focus critique on design decisions, not UI polish. The question is: does this application give the user the information they need, in the right order, with the right navigation?

**Assessment:** Not formally evaluated — this is a formative design exercise. Instructor notes design strengths and gaps for Day 3 Ontology Design exercise awareness.

---

## DAY 2 — ADVANCED PIPELINE BUILDER

### Block 5 — Multi-Source Joins

---

**Hours:** 2.0 | **Method:** Lab | **Time:** 0830–1030

**Purpose:** Production pipelines typically join 3+ sources. Fan-out from a many-to-many join is the most common and most destructive data quality error. Mastery of join type selection and fan-out detection is non-negotiable at SL 3.

**TLO:** Given three datasets with different join conditions, the trainee will build pipelines using inner, left, and outer joins, identify fan-out in a provided example, correct it, and document the resolution.

**Key Delivery Notes:** - Inner/left/outer join review: build all three on the same dataset and compare row counts — the difference is instructive. - Fan-out detection: provide a dataset with a known 1:N relationship. Show the row multiplication. Check with: "expected row count after join" = left-side row count (inner join) or right-side row count (outer) — if it's higher, there's fan-out. - Fan-out correction: deduplicate the right-side lookup table before joining, or use a GROUP BY to collapse multiples. - Document the resolution in a pipeline step comment — "deduplication added to address fan-out from unit\_history table".

**Assessment:** Evaluated in Practical Exercise Task 2.

---

## Block 6 — Union Transforms

---

**Hours:** 1.25 | **Method:** Lab | **Time:** 1045–1200

**Purpose:** Union operations combine datasets from multiple sources into a single dataset — essential for cross-unit or cross-period analysis. Schema mismatch handling is the key skill.

**TLO:** The trainee will build a union pipeline combining two datasets with compatible schemas and correct a schema mismatch using a Rename/CAST step before the union.

**Key Delivery Notes:** - Union requires matching column names and types in both inputs. Foundry will error on type mismatch and silently null on name mismatch (column exists in one, not the other). - Correction pattern: add a Rename or CAST step before the union on the non-conforming dataset. Align to the "canonical" schema, not to whatever the source provides.

**Assessment:** Practical Exercise Task 2 (multi-source pipeline).

---

## Block 7 — Group-By Aggregations

---

**Hours:** 2.0 | **Method:** Lab | **Time:** 1300–1500

**Purpose:** Aggregation from detail to summary is essential for portfolio dashboards and commander-level products. The aggregate-then-join pattern is the key technique to master.

**TLO:** Given a detail dataset, the trainee will build group-by aggregation steps computing count, sum, and computed columns — then join the aggregated result back to a dimension dataset.

**Key Delivery Notes:** - Aggregate-then-join pattern: aggregate raw data to unit level (count, sum), then join to a unit lookup table. This is a two-step pipeline — aggregate first, then join. Joining raw data before aggregating often causes fan-out. - Multiple aggregations in one step: `COUNT(equipment_id) AS total_equipment`, `SUM(CASE WHEN c_rating = 'C1' THEN 1 ELSE 0 END) AS c1_count`. - Computed aggregation columns: `c1_count / total_equipment * 100 AS c1_rate`. Computed after the aggregation step using a Calculated Column step.

**Assessment:** Practical Exercise Task 2.

---

## Block 8 — Append Mode and Snapshot Pipelines

---

**Hours:** 1.25 | **Method:** Lab | **Time:** 1515–1630

**Purpose:** Append mode enables historical records — essential for trend analysis and audit trails. Configuring it incorrectly is the most common SL 3 data governance error.

**TLO:** The trainee will configure a pipeline in Append mode, run it twice, verify two distinct snapshot records exist with different timestamps, and document when Append vs. Overwrite mode is appropriate.

**Key Delivery Notes:** - Add a timestamp column to the pipeline output before the Append run: `CURRENT_TIMESTAMP() AS snapshot_date`. This distinguishes snapshot records. - Configure Append mode BEFORE the first run. If you run in Overwrite first, your second Append run produces 2 records (correct) but you've already lost the ability to confirm it was configured correctly from the start. - Document the mode choice in a pipeline step comment.

**Assessment:** Evaluated in Practical Exercise Task 3.

---

## Block 9 — Scheduled Pipeline Configuration

---

**Hours:** 0.5 | **Method:** Lab | **Time:** 1630–1700

**Purpose:** Production pipelines run on schedule. Trainees must be able to configure a schedule and a failure alert — a pipeline that runs silently and fails with no notification is worse than no pipeline.

**TLO:** The trainee will configure a pipeline schedule expression and a build-failure email alert — verifying both in the pipeline settings.

**Key Delivery Notes:** - Schedule expression: cron-style (`0 6 * * 1-5` = 0600 Monday–Friday). Confirm the expression matches the intended schedule. - Build failure alert: email to the pipeline owner (minimum) and the data steward.

---

## DAY 3 — ONTOLOGY DESIGN

### Block 10 — Ontology Design Methodology (Lecture)

---

**Hours:** 1.0 | **Method:** Lecture | **Time:** 0900–1000

**Purpose:** Good Ontology design comes from understanding the domain before touching the tool. This lecture establishes the SL 3 design methodology: entity identification, relationship mapping, Action design — all on paper before Ontology Manager opens.

**TLO:** The trainee will describe the 4-step SL 3 Ontology design methodology and apply it to a provided domain description — producing a documented design before any tool interaction.

**Key Delivery Notes:** - Step 1: Domain entities — what are the "things" in this domain? Equipment, Units, Milestones, Personnel, Documents. - Step 2: Properties — what do we need to know about each thing? What does a user query on? - Step 3: Relationships — which entities relate to which? What is the cardinality? - Step 4: Actions — what do authorized users need to do? What properties do they write? Who is authorized? - Every SL 3 design must be documented before it is built. "I'll figure it out as I go" fails the design review.

---

### Block 11 — Individual Design Exercise

---

**Hours:** 1.75 | **Method:** Lab | **Time:** 1015–1200

**Purpose:** Applying the design methodology independently to a provided mission scenario. The product is a documented Ontology schema reviewed against the 6-item rubric before building is permitted.

**TLO:** Given a mission scenario, the trainee will produce a documented Ontology schema (Object Types, properties, Link Types, cardinality, Actions) reviewed against the design rubric — achieving  $\geq 75\%$  rubric score before the build phase begins.

**Key Delivery Notes:** - Provide a mission scenario with enough ambiguity to require design decisions (e.g., "A unit needs to track equipment, maintenance status, and assignment to crews"). - Trainees document the design on paper or in a structured template — NOT in Ontology Manager. - Instructor reviews each trainee's design against the rubric before allowing them to proceed to Block 13. A design with fatal flaws is corrected here, not after objects are created. - Design rubric items: (1) entities correctly identified, (2) Primary Keys chosen appropriately, (3) property types correct and documented, (4) Link Type cardinality correct, (5) Actions target correct properties with appropriate access control, (6) Naming follows C2DAO convention.

**Assessment:** Design review is a prerequisite to Block 13. Go/No-Go decision point.

---

## Block 12 — Design Critique Workshop

---

**Hours:** 2.0 | **Method:** Workshop | **Time:** 1300–1500

**Purpose:** SL 3 builders are the data leads of their formation. They must be able to explain and defend design decisions. The critique session builds this professional skill.

**TLO:** Each trainee will present their Ontology design to the class, receive critique from peers against the design rubric, and revise the design as directed before build.

**Key Delivery Notes:** - Each trainee presents their design (2–3 minutes). Class applies the rubric. Instructor facilitates. - Common rubric failures at SL 3: cardinality without justification, duplicate Object Types for entities that should be the same type, Actions with no access restriction. - Critique is professional, not personal. "The cardinality on Equipment-to-Unit is MANY\_TO\_MANY but the scenario implies MANY\_TO\_ONE — what is the reason for that choice?"

---

## Block 13 — Build the Approved Design

---

**Hours:** 2.25 | **Method:** Lab | **Time:** 1515–1700

**Purpose:** Build the design approved in Block 12. Trainees who skip design and go straight to the tool make this mistake: they rebuild when the design is wrong. Trainees who designed first build quickly and correctly.

**TLO:** Given an instructor-approved Ontology design, the trainee will implement it in Ontology Manager and connect it to a pipeline via an Ontology write step — Object count in Quiver matches expected.

---

# DAY 4 — ANALYTICS TOOLS AND AIP LOGIC

## Block 14 — Contour: Advanced Analysis

---

**Hours:** 2.0 | **Method:** Lab | **Time:** 0830–1030

**Purpose:** SL 3 Contour use goes beyond basic charts. Pivot tables, calculated columns, and saved workbooks turn Contour into a repeatable analysis tool rather than a one-time chart builder.

**TLO:** The trainee will build a Contour workbook with a pivot table, a calculated analysis column, a parameter control, and a saved view shared with the project.

**Key Delivery Notes:** - Pivot tables in Contour: row dimension, column dimension, value aggregation. Behavior differs from Excel — no drag to column, all configuration is in the field panel. - Calculated columns: Contour allows computed columns directly in the analysis view without modifying the underlying

dataset. Useful for deviation-from-standard calculations. - Parameter controls: interactive input that the analysis user can adjust (e.g., "threshold" slider that changes the highlighting threshold in the chart). Configured in Contour view settings.

---

## Block 15 — Quiver: Multi-Object Analysis

---

**Hours:** 1.25 | **Method:** Lab | **Time:** 1045–1200

**Purpose:** Quiver's power is in linked views and cross-filter propagation — multiple Object Type views that respond to a selection in one view. This is the SL 3 skill that SL 1 trainees cannot do.

**TLO:** The trainee will configure a multi-Object Quiver dashboard with linked views and cross-filter propagation — selecting an Object in one view narrows the related objects shown in a second view.

**Key Delivery Notes:** - Add multiple Object Type views in the same Quiver workspace. - Configure the link: selecting a Unit Object in View 1 filters Equipment Objects in View 2 to only show Equipment linked to that Unit. - The link must be explicitly configured — it does not happen automatically. This is the most common practical exercise failure point in Quiver.

---

## Block 16 — AIP Logic Configuration

---

**Hours:** 1.5 | **Method:** Lab | **Time:** 1300–1430

**Purpose:** SL 3 covers AIP Logic configuration — connecting triggers, inputs, and outputs — not authoring. Trainees learn to configure and review workflows, not build them from scratch.

**TLO:** Given an existing AIP Logic workflow template, the trainee will configure input sources, output destinations, trigger conditions, and a human review queue — verifying the workflow routes uncertain outputs to review before writing to production Objects.

**Key Delivery Notes:** - Configuration vs. authoring: SL 3 trainees configure pre-built workflows (changing input source, output Object Type, trigger). They do not author the prompt logic — that is SL 4H. - Human review queue design: all AIP Logic outputs that write to production Objects must route through a review queue. Configuration: output with status = "Draft" → human review → status = "Published". No direct writes to Published status. - Trainees should be able to read the lineage graph and identify what the workflow does before configuring it.

---

## Block 17 — Data Lineage

---

**Hours:** 1.25 | **Method:** Lab | **Time:** 1445–1600

**Purpose:** Lineage graphs are how SL 3 builders diagnose pipeline failures and understand upstream data dependencies. A builder who cannot read a lineage graph cannot effectively maintain production data products.

**TLO:** Given a Foundry lineage graph, the trainee will identify the upstream source datasets, the transforms in the pipeline, and the downstream consumers — and use this to diagnose a described pipeline issue.

**Key Delivery Notes:** - Open lineage view from any dataset or pipeline. Show upstream (left) and downstream (right). - Diagnosis exercise: instructor describes a pipeline failure. Trainees use the lineage graph to identify which upstream dataset likely caused it. - Real-world use: "My Workshop application stopped updating." → Lineage graph shows the pipeline feeding it → Pipeline shows a failed step → That step reads from an upstream source that hasn't refreshed. Fix is at the upstream source, not in the pipeline.

---

## Block 18 — C2DAO Production Standards Discussion

---

**Hours:** 1.0 | **Method:** Discussion | **Time:** 1600–1700

**Purpose:** Production standards are what separate a training exercise from a data product a commander can trust. This discussion establishes the professional standard for SL 3 builders.

**Key Delivery Notes:** - What makes a data product production-ready: correct naming, classification marking, documented Ontology design, pipeline running without error on schedule, workshop application tested with correct access control, data steward approval. - The difference between "it works" and "it's production-ready" — trainees have seen both in their labs this week. Instructor identifies examples.

---

## DAY 5 — GOVERNANCE AND PRACTICAL EXERCISE

### Block 19 — Full C2DAO Promotion Workflow

---

**Hours:** 1.0 | **Method:** Lab | **Time:** 0800–0900

**TLO:** The trainee will execute the full C2DAO promotion workflow: create branch → make change → write complete description → submit → respond to evaluator feedback → confirmation.

**Key Delivery Notes:** - Evaluator plays data steward. They will reject the first submission if the description is generic. This is intentional. - Complete description format: "What changed / Why / Downstream impact / Author / Date."

---

## Block 20 — Full-Stack Review

**Hours:** 1.0 | **Method:** Review | **Time:** 0900–1000

**Purpose:** Self-assessment before the evaluation. Each trainee traces their own data product end-to-end and identifies production-readiness gaps.

## Block 21 — Practical Exercise Scenario Brief and Design Time

**Hours:** 1.25 | **Method:** Brief | **Time:** 1015–1100

**Key Delivery Notes:** - Provide scenario brief. Trainees have 30 minutes of planning time — they must document their Ontology design before the evaluation begins. - Evaluator reviews the design document before permitting the build phase. A fatally flawed design is flagged and must be corrected.

## Block 22 — Practical Exercise (Evaluated)

**Hours:** 4.0 | **Method:** Evaluation | **Time:** 1300–1700

**Scenario:** S3 requires a multi-source readiness dashboard combining personnel and equipment data, battalion portfolio view with drill-down to unit detail.

**Tasks:** (1) Design and document Ontology schema — evaluated against design rubric before build. (2) Build multi-source join pipeline; compute readiness score per unit. (3) Configure Append mode; run twice; verify two snapshot records. (4) Build multi-page Workshop: portfolio → unit detail with variable passing. (5) Build Contour workbook with calculated deviation column. (6) Submit pipeline and application through C2DAO promotion workflow; respond to steward feedback.

**Go standard:** All 6 tasks. Design rubric  $\geq 75\%$  with no zero-score item. Hard No-Go: promotion submitted without description; data steward feedback not responded to.

## COURSE COMPLETION — NEXT STEPS

Upon receiving a SL 3 Go result, trainees are eligible to enroll in any SL 4 track — all six WFF functional tracks (SL 4A–F) and all eight specialist tracks (SL 4G–O). The specialist tracks are:

Track	Title	Prereq	Duration
SL 4G	ORSA Specialist	SL 3 Required	5 days

Track	Title	Prereq	Duration
SL 4H	AI Engineer	SL 3 Required	5 days
SL 4M	ML Engineer	SL 3 Required	5 days
SL 4J	Program Manager	SL 3 Required	4 days
SL 4K	Knowledge Manager	SL 3 Required	4 days
SL 4L	Software Engineer	SL 3 Required	5 days
SL 4N	UI/UX Designer	SL 3 Required	5 days
SL 4O	Platform Engineer	SL 3 Required	5 days

**SL 3 is a hard prerequisite for all specialist tracks — not a recommendation.** Enrollment requests missing SL 3 Go documentation will not be confirmed by C2DAO.

#### NOTE

Upon completing SL 3, trainees are eligible for ALL SL 4 tracks (A–O) — both WFF functional (A–F) and specialist (G–O). SL 3 is a hard prerequisite for every SL 4 track.

**T3-I (Instructor Certification) — parallel path:** SL 3 graduates who demonstrate strong instructional aptitude may be nominated for T3-I (Instructor Certification). T3-I is a 5-day course that qualifies graduates to deliver SL 1, SL 2, and SL 3 as certified instructors. Nominations require unit commander endorsement and C2DAO approval.

Instructors should discuss which specialist track is appropriate for each trainee at the end of Day 5. For trainees with strong instructional aptitude, also highlight T3-I as an available path. Refer trainees to the Unit Training NCO/Officer for enrollment coordination and access provisioning lead times (7–10+ duty days for specialist tracks).

---

USAREUR-AF Operational Data Team SL 3 Lesson Plan Outlines | Version 1.0 | March 2026