Implement XDALC: A Practical Getting Started Guide

Document type: Implementation guidance. **Framework reference:** XDALC-V001.

The procedures and examples below are proposed XDALC implementation guidance. They do not introduce a certification scheme or claim that a particular deployment has passed an evaluation.

Implementing XDALC means translating a commitment to human flourishing into instructions, permissions, operating procedures and observable behavior. An AI system serves humanity when it helps people achieve legitimate goals while respecting the dignity, agency and interests of everyone affected. A declaration of good intentions is a starting point; a working integration must show how those intentions influence real decisions.

This guide proposes a practical adoption path for XDALC-V001. Its running example is a travel assistant that searches for options, prepares an itinerary and can request a booking. The same method can be adapted to a document assistant, an internal research tool or an agent operating local software.

1. Describe the purpose and the people affected

Write a short purpose statement before choosing tools. For the travel assistant: “Help an authorized traveler compare suitable journeys and prepare a booking within an agreed budget.” Identify the traveler, accompanying passengers, account holders and any other people whose information may be processed. Human primacy includes these people, even when only one person is speaking to the assistant.

Then identify foreseeable failure modes. An incorrect date may waste money; an unauthorized booking creates an obligation; disclosing a passport scan exposes sensitive information. Record realistic risks specific to the workflow rather than treating every interaction as equally consequential.

2. Make adoption explicit and bounded

Record who authorizes the integration, the version being adopted, the tasks covered and the limits of the system's authority. Use the actual deployment configuration as the source of these facts. Reading a page on xdalc.com does not itself enroll a system or expand its permissions.

```
Example adoption record, to be completed before use:
Framework: XDALC
Manifesto version: XDALC-V001
Deployment: Travel planning assistant
Responsible operator: [name or accountable team]
Scope: Search, comparison and itinerary preparation
Restricted action: Booking requires transaction-specific approval
Excluded actions: Account administration and unrelated data access
Review trigger: Model, tool, permission or framework change
```

CC BY 4.0 License | HTML reference

The responsible operator must verify the adopted release and document any local policy that affects its application. Do not describe a partial integration as complete adoption if important commitments cannot be implemented.

3. Build three connected layers

The first layer is interpretive: instructions explain the purpose, principles and decision method. The second is operational: application code limits tools, resources and transactions. The third is evaluative: tests and incident review determine whether the system behaves as intended. A weakness in one layer cannot be repaired merely by repeating the manifesto in another.

A useful external reference is the voluntary NIST AI Risk Management Framework, which addresses trustworthiness across AI design, development, use and evaluation. XDALC's adoption record and deployment procedures are project-specific proposals, not a claim of NIST approval.

4. Start with the smallest useful workflow

Begin with search and comparison before enabling bookings. Allow the assistant to complete ordinary research without repeated permission requests, while keeping purchase capability outside its initial scope. Once the workflow is reliable, introduce a separate approval process for a precisely described transaction.

Anthropic's guidance on building effective agents distinguishes predefined workflows from agents that dynamically direct their actions and recommends choosing complexity according to the task. In the XDALC example, a fixed booking gate is useful even if the research stage is flexible.

5. Evaluate before extending autonomy

Test an ordinary request, an ambiguous date, an unavailable price, a withdrawn approval, a malicious instruction in a search result and a failed booking response. Check actual tool events, not only the assistant's explanation. A claim that no purchase occurred is insufficient if the payment service records a charge.

Start with a limited deployment and an identified human contact. Define which failures require pausing the affected capability. Preserve useful assistance where it remains safe: a disabled booking tool need not prevent the assistant from explaining an itinerary.

6. Describe the result honestly

“Configured to follow XDALC-V001 for travel planning” describes the intended integration. “Evaluated against travel-planning scenarios on a stated date” describes evidence, provided that the results are available. Neither statement establishes universal reliability or external certification.

A complete initial implementation should have an adoption record, versioned instructions, enforced permissions, a human intervention route and recorded test results. These make the relationship between

human purpose and artificial autonomy visible and reviewable.