Build an AI Assistant for Your Progress OpenEdge Application

A Practical Path Using the Progress Agentic RAG Solution

WHITEPAPER

This paper outlines a practical approach for Progress® OpenEdge® Independent Software Vendors (ISVs) to embed AI assistants directly into their applications using the Progress® Agentic RAG solution. Its core argument is that the most effective AI strategy is not to replace trusted business systems with generic chat tools, but to extend existing applications with assistants that operate within established workflows, use governed application data and respect existing business logic and permissions. By combining structured OpenEdge data, unstructured business knowledge, the OpenEdge Model Context Protocol (MCP) Server and a language model grounded in authoritative sources, ISVs can deliver assistants that explain business context, reduce manual effort and support faster decisions.

It also emphasizes starting with a narrow, high-value use case—especially where users must currently reconcile information across multiple systems—and then expanding incrementally over time. This approach helps ISVs reduce implementation risk while improving product differentiation, monetization opportunities, customer retention and the overall value of their applications.

Introduction

Progress OpenEdge ISVs have spent years building applications that customers rely on to run critical parts of their business. These systems support finance, operations, supply chains, service delivery and other core processes every day. They are stable, trusted and deeply embedded in how organizations work.

Customers increasingly expect their software to do more than store and display data. They expect it to help them understand what is happening, explain why it is happening and guide the next step, directly within the workflows they already use.

An embedded AI assistant delivers on that expectation, but only when it goes beyond a generic chat interface. The goal is not to add another way to ask questions, but to solve specific business problems inside the application, using the same data, rules and processes the business already trusts.

The Progress Agentic RAG solution provides a practical way to build this capability. It allows ISVs to start with a focused use case, connect the right data and knowledge and deliver an assistant that produces actionable, business-aware responses, not just conversational answers.

Why an Embedded Assistant Makes Sense

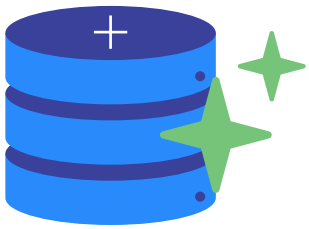
Many AI initiatives struggle because they begin too broadly or remain detached from the product themselves. They require significant upfront investment and configuration before users see tangible value, and they often operate outside the core application, which can limit adoption.

An embedded assistant takes a more pragmatic path. It starts with a specific problem users already encounter, such as understanding data, answering common questions or navigating complex workflows. By addressing a defined use case inside the application, ISVs can deliver visible value early while maintaining control over scope and risk.

Because the assistant is integrated into the application experience, users do not need to learn a new system. Interactions occur in context, using the same data, permissions and rules the application already enforces. For ISVs, this approach reduces disruption and supports a more incremental adoption of AI capabilities over time.

This model is fundamentally different from embedding a generic chat interface. A standalone chat experience can respond to questions, but it does not understand application state, enforce business rules or reliably access governed data.

An embedded assistant is designed around specific workflows and decisions. It operates on trusted application data, respects permissions and uses existing services to execute logic. The result is not just an answer. It is a response that reflects how the business actually operates and can be used immediately.



Architecture Overview: How an OpenEdge Assistant Works

An AI assistant in an OpenEdge environment is not a standalone system. It is an extension of the existing application architecture.

At the user level, the assistant typically appears as a contextual panel, command interface or assistant embedded within the application UI. Users interact using natural language, asking questions or requesting help related to their current task.

Behind the scenes, several components work together:

- **Progress Agentic RAG** solution acts as an orchestration layer. It interprets user requests, determines what information is needed and coordinates retrieval and response generation.
- **Retrieval sources** include structured OpenEdge data (such as transactions, customers or operational records) and unstructured content (such as documentation, knowledge articles or support notes). Which sources are included depends on how the use case is designed.
- **The OpenEdge MCP Server** provides a governed way for the AI workflow to interact with the application. Rather than bypassing business logic, the assistant uses the same APIs and services already exposed by the product. This helps enable the consistency of permissions, validations and rules.
- **A language model** supports natural-language interaction and response generation. Importantly, it is not treated as the system of record, but grounded in application data and logic.

This architecture allows the assistant to enhance the application without changing its fundamental behavior or integrity.

From Concept to First Assistant Use Case

Building an embedded assistant does not start with a generic chat interface. It starts by selecting a specific workflow where users currently can't get a complete answer from a single source: one that requires navigating multiple systems, interpreting documents alongside data or relying on experienced colleagues to fill the gaps.

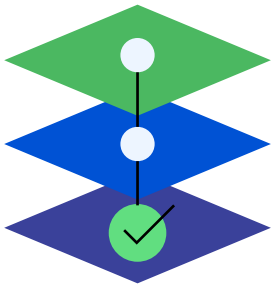
For OpenEdge ISVs, the strongest starting points are questions that already cross the boundary between structured and unstructured knowledge. For example:

- Questions users frequently ask support or professional services, such as why a pricing override occurred or whether a customer is at compliance risk
- Reports users run regularly, but struggle to interpret without consulting policies, contracts or historical context that lives outside the database
- Workflow steps that force users to access OpenEdge transaction data, then manually cross-reference documentation, approvals or case history to reach a conclusion

The clearest signal that a use case is right for an assistant is the swivel-chair pattern: a user who must open multiple systems, reconcile what they find and apply their own judgment before they can answer a question that should be answerable in seconds. That gap between what the system stores and what the user needs to know is **exactly where agentic retrieval-augmented generation (RAG) technology operates.**

Once a use case is selected, implementation typically follows a clear progression:

1. The ISV defines which application data and content sources are required to answer that question accurately. This may include OpenEdge tables, business logic exposed through services and supporting content such as policies, contracts or operational documentation. Mapping both sides of that divide upfront facilitates sufficient context of the retrieval layer to produce grounded answers.
2. The Progress Agentic RAG solution is configured to ingest, enrich and index those sources so they become queryable as a unified knowledge foundation. When a user submits a question, the solution interprets the query and retrieves the most appropriate context from across those sources. Where the answer depends on live application data, the agent invokes defined services through the OpenEdge MCP Server rather than querying raw tables directly, keeping business logic centralized and security models intact. The retrieved content is then refined and ranked before being passed to the language model, so the generated response is grounded in authoritative sources rather than approximation.



Finally, a lightweight assistant interface is embedded into the application at the point where users already work. Rather than introducing a freeform chat experience, the assistant is scoped to the workflow it supports—surfacing answers in context, with citations that allow users to verify the source of any claim.

This approach keeps the initial implementation focused and delivers measurable value quickly while creating a foundation that can be expanded to additional workflows over time. Each use case deepens the system's value without requiring changes to the underlying application architecture.

Grounding Responses in Application Data and Knowledge

The usefulness of an assistant depends heavily on what it can access and how that access is controlled.

OpenEdge applications already contain data, process logic and domain expertise that AI can help surface more effectively. The Progress Agentic RAG solution connects to these sources to retrieve relevant information and assemble responses that reflect how the business actually operates.

For example, a user might ask why certain customers are paying invoices late. Depending on how the assistant is configured, it could retrieve payment history, reference relevant contractual terms or policies, synthesize this information and return a summarized explanation. In some cases, it may also suggest potential next steps, based on predefined rules or historical patterns available to the application.

This differs from traditional reporting which presents raw data and requires interpretation, instead of an assistant helping users understand what the data means while still relying on authoritative sources within the application.

Because responses are grounded in the same data and logic the application already uses, results remain consistent with existing behavior.

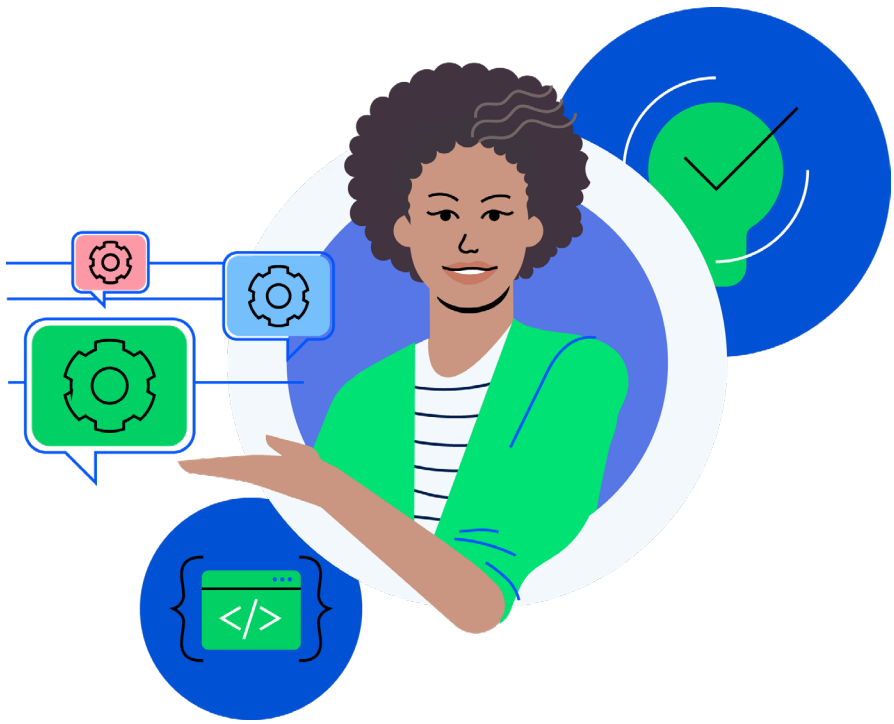
Using AI Inside Everyday Application Workflows

The value of an embedded assistant becomes clear when it is used in context.

Consider a user working in an OpenEdge-based ERP system. Today, answering a question about accounts receivable might involve running reports, applying filters and manually interpreting results. With an embedded assistant, that user can ask a question in natural language and receive a response that summarizes relevant information and explains it in plain terms.

The interaction stays within the application. Users do not need to rely on separate tools or export data to external systems. Over time, this can reduce friction, make complex systems easier to use and help users make decisions more efficiently.

Across different types of OpenEdge applications, similar patterns can apply. In financial systems, an assistant may help explain trends or anomalies. In supply chain or operations software, it may assist with identifying delays or bottlenecks. In service environments, it may summarize cases or highlight relevant history. Each use case builds on the application's existing strengths.



Solving Business Problems, Not Just Answering Questions

The value of an embedded assistant is measured by the **business problems it removes, not by how well it can converse.**

Below are examples of how it delivers measurable impact:

Finance (Accounts Receivable)

- **Today:** A user runs multiple reports, reconciles inconsistencies and manually interprets results
- **With an assistant:** The system analyzes payment history, applies contractual rules and explains *why* invoices are late—highlighting risk factors and next steps
- **Impact:** Faster resolution, reduced manual investigation and improved cash flow visibility

Operations/Supply Chain

- **Today:** I dentifying delays requires navigating multiple screens and interpreting fragmented data
- **With an assistant:** The system correlates transaction data, inventory constraints and supplier timelines to explain the root cause of a delay
- **Impact:** Faster issue resolution, reduced operational downtime and better decision-making

Service Management

- **Today:** Agents review long histories and search knowledge bases before responding
- **With an assistant:** The system summarizes prior interactions, surfaces similar cases and highlights relevant guidance
- **Impact:** Shorter handling time, more consistent responses and improved customer experience

In each case, the assistant is not just answering a question. It is **reducing time to insight and removing manual effort** from critical workflows.

Delivering Value Incrementally

Introducing an AI assistant does not require a complete redesign of an OpenEdge application. ISVs can begin with a narrowly defined use case that delivers clear value, such as answering common questions, simplifying access to key data or providing contextual guidance in a specific workflow. From there:

- Relevant data and content sources are connected through the OpenEdge MCP Server
- Agentic RAG technology is configured to retrieve and assemble responses for that use case
- The assistant interface is embedded into the application experience

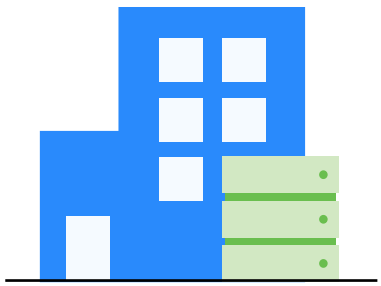
Because the core system remains unchanged, this approach keeps risk manageable. Once the initial scenario is in production, additional use cases can be added over time, expanding the copilot's capabilities as confidence and value grow.

For some ISVs, a focused initial implementation can be delivered in a relatively short timeframe. However, timelines depend on factors such as data quality, integration readiness and the scope of the use case.

How an Embedded Assistant Is Built (in Practice)

At a high level, building an embedded assistant is not about training a model or adding chat. It is about connecting three things that already exist in the application:

1. **Business data** (transactions, customers, operational records)
2. **Business knowledge** (policies, contracts, documentation, case history)
3. **Business logic** (workflows, rules, permissions)



The Progress Agentic RAG solution brings these elements together through the structured flow as follows:

- A user asks a question within a specific workflow
- The system identifies what data and knowledge are required to answer it
- Relevant information is retrieved from both structured and unstructured sources
- Application services are invoked where business logic is required
- A grounded response is generated with traceable sources

This is what differentiates an embedded assistant from a chat experience. The system is not generating answers based on general knowledge. *It is assembling answers from trusted sources and application logic*, so users can act on them immediately.

Business Impact for ISVs

For OpenEdge ISVs, an embedded assistant creates business impact that goes beyond improving user experience. It becomes a product capability that drives revenue, differentiation and retention due to:

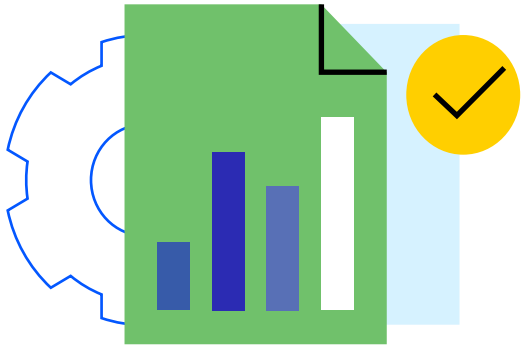
- **Market Positioning:** AI is no longer optional in competitive evaluations. An embedded assistant provides a visible, credible way to demonstrate innovation tied directly to real workflows, not as a standalone feature. Relevant information is retrieved from both structured and unstructured sources
- **Monetization:** AI capabilities can be packaged as premium features, usage-based services or new product tiers. More importantly, they allow ISVs to tie pricing to outcomes and productivity gains, not just access to software.
- **Retention and Expansion:** When software actively helps users complete tasks faster and make better decisions, it becomes harder to replace. The application moves from a system of record to a system users depend on for insight, strengthening long-term customer value.

Conclusion

OpenEdge ISVs already possess many of the elements required to deliver meaningful AI experiences: trusted data, established workflows and deep domain expertise.

The Progress Agentic RAG solution provides a way to unlock that value in a controlled, practical manner. By embedding AI into an OpenEdge application, ISVs can enhance usability, support better decision-making and evolve their products without disrupting what already works and grow their business and profitability.

The application remains the system of record, while increasingly becoming a system of insight.



Learn about the Progress Agentic RAG for OpenEdge solution

About Progress Software

Progress Software (Nasdaq: PRGS) empowers organizations to achieve transformational success in the face of disruptive change. Our software enables our customers to develop, deploy and manage responsible AI-powered applications and personalized digital experiences with agility and ease. Businesses of all sizes get a trusted provider in Progress, with the products, expertise and vision they need to turn AI disruption into a competitive advantage. Millions of developers and technologists at hundreds of thousands of organizations depend on Progress every day. Learn more at www.progress.com

Worldwide Headquarters

Progress Software Corporation
15 Wayside Rd, Suite 400, Burlington, MA 01803, USA
Tel: +1-800-477-6473

facebook.com/progresssw
x.com/progresssw
youtube.com/progresssw
linkedin.com/company/progress-software
progress_sw_

Progress