

Hand Coding vs. Data Integration Tools. Which Should You Choose?

The Pros and Cons of Each Approach—
and 10 Questions to Ask Before Making a Decision

With custom coding, there is no guaranteed consistency from one developer to the next, making development and maintenance complex and costly.

Many data professionals ask themselves why they need a data integration tool when hand coding can often do the job quickly and at a lower upfront cost.

The true cost of hand coding is not initially understood. Despite this lack of clarity, many companies still manage integration projects by writing custom code—typically to satisfy development efforts driven by an immediate business need. The upfront benefits seem compelling: short start time, no new development tools, and simple deployment. However, as integration requirements inevitably increase, problems with custom code will begin to mount.

Custom code can't be reused, can be challenging to maintain, and often, the original developer is no longer available. Custom solutions are neither scalable nor extensible. As a result, what started as a simple project often becomes far more complex and costlier than initially expected.

Every company must consider numerous factors when evaluating the tradeoffs between hand coding and a tool-based approach to data integration. Hand coding and data integration tools are often compatible but understanding when to use each method can be difficult to determine. Most companies use a combination when tackling their technical challenges.

Information management infrastructure architects and lead implementers must always choose between using tool-based or custom-code development for data integration. Both have their place.

When considering custom-coded projects, IT decision makers must ask themselves these 10 questions

1. What are the short- and long-term costs of the initiative?

It's important to understand that maintenance and support costs are part of any project. While initial deployment costs might be reduced by 20% with a custom-coded approach, the maintenance costs will increase by 200%. A data integration tool may be a better choice if you want to build a repeatable, dependable, and sustainable process.

If different people maintain and support the code once it's in production, they will face an extensive learning curve with a hand-coded approach. Even worse, if the code is in production for years and the person who developed it leaves the company, then understanding how the integration job works—and how to fix it when it breaks—becomes exponentially more complex and more expensive.

2. Does my development team have the expertise to do this project using hand coding?

If you're using a new technology, such as a cloud platform, who will do the work, and how much ramp time will they need? Organizations often fail to balance the availability of mature staff against the demand for innovation or expert implementation requirements. Thus, they fail to realize that those tool-based and custom-coded solutions can only be delivered according to the ability and availability of their staff. This often means highly skilled staff with direct experience with the exact code and requirements are needed.

3. Is this how I want my hand-coding experts to spend their time?

Hand-coding experts typically make up about a quarter of the entire development team, making them a scarce and expensive resource. If a non-expert could do the same work using a tool—and save hours of time—then wouldn't you rather have the experts do work that requires their unique skills?

4. Can I do this same work with a tool cheaper and faster than my team can hand code?

Most IT teams are constantly being asked to do more with less. Compared to hand coding, a tool-based approach often allows a lower cost per developer to do the work and accomplish it quickly.

5. Is this a one-off, stand-alone project, or an area where I plan increasingly more development?

There is a time and place for hand coding, but only in particular situations. Custom coding can make sense for targeted, simple, and one-off projects that do not require much maintenance. It can also be necessary when no tools can do the required work. If you embark on an initiative using a big data or cloud platform, you will likely want to do more work on that platform. If so, relying on expert hand coders will be a tough approach to scale, given the scarcity of these resources.

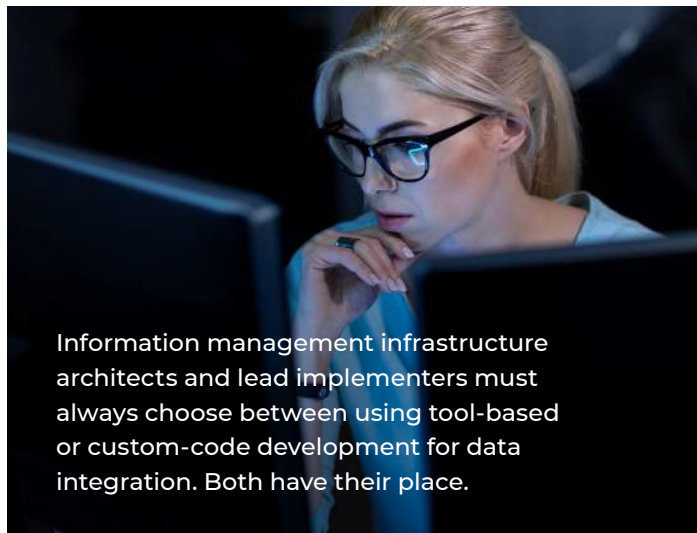
6. How portable will this code be if I want to repurpose it on a new technology platform, such as Spark or Flink?

When budgeting costs and time for an integration project, think about the additional efforts needed to redevelop all previous work and the efforts needed for new development. Leading data integration tools allow you to move from one data processing framework to another, eliminating legacy code situations.

7. Will multiple developers collaborate on this project?

Data-integration projects requiring multiple developers will benefit from the visual design environments provided by tools—easy reuse and code sharing, automated documentation, and even wizards and experts to advise the developer. If several developers are working on the integration initiative, it is essential to consider how the developers' work will fit together.

With custom coding, there is no guaranteed consistency from one developer to the next, making development and maintenance complex and costly. A solution that can reuse prior development elements will keep your integration team from duplicating efforts and result in more efficient data integration flows. It's often easier to write code than it is to read it.



Information management infrastructure architects and lead implementers must always choose between using tool-based or custom-code development for data integration. Both have their place.

8. How long will this code be in production?

When embarking on a new project, it's tempting to focus on the time needed to develop it and forget how long it will be in production. Often, a project that requires six months to develop will be in production for five years or longer. If that is the case, then that code's support and maintenance costs will continue for 10 times longer than the initial development work, so it's critical you understand your support and maintenance costs.





9. Who will own the maintenance of this code?

If you have only a handful of developers, then they will be forced to maintain and support their code. Eventually, support and maintenance will consume all of their capacity, making it impossible for them to start new projects that could help your organization gain a competitive edge. It's important to understand the maintenance and support costs of any project. If different people maintain and support the code once it's in production, they will likely face an extensive learning curve with a hand-coded approach, and if the code is in production for years, turnover will lead to much larger costs.

10. How often will the code need to be updated?

The code may need to be updated to accommodate new business needs or changes in data sources or targets. Data sources, targets, and business needs are constantly evolving. If it's reasonable to expect this constant stream of changes, then the cost of maintenance and support will be significantly more.

Hand Coding vs. Integration Solution

Category	Hand Coding	Integration Solution
	Very large. Constantly requires skilled development resources.	Very small. Easily maintain integration with a few clicks.
 Reusability	Little to none. Each integration must be built from scratch. Little standardization between integration projects.	Yes. Designed with reusability as one of its core tenets, with workflow templates to accelerate the integration projects.
 Monitoring and Error Detection	Manually build monitoring and auditing tools to manage error handling and logging functions.	Automatically monitor, detect, and notify of errors using built-in tools.
 API Restrictions	Introduces significant complexity, which can hamper the integration performance process and scalability.	Processes data faster and delivers an optimized solution without the pain of iterative development.
 Changes to Business Processes and APIs	Very expensive and time consuming to manage, requiring skilled programmers.	Dynamic configuration changes to cope with business process changes and APIs.
 ROI	Variable. Small upfront costs heavily offset by constant maintenance needs, lack of reusability, and frequent integration breaks that compromise application value.	Large ROI. Significant time and resources are saved during implementation, execution, and maintenance of an integration project.

Learn more

To learn more about how Actian integration solutions can jump-start your data integration project, email: DataConnect.Enterprise.Sales@actian.com to request a no-cost evaluation license or visit our website at [Actian.com](https://actian.com).

About Actian

Actian transforms business by simplifying how people connect, manage, and analyze data. The Avalanche Cloud Data Platform gives customers a flexible and comprehensive solution that enables fast innovation by gaining real-time insights, leveraging native data integration, and deploying anywhere. The Actian platform offers maximum interoperability by combining data integration, management, and analytics solutions across systems, while its partnership model makes scaling fast, efficient, and effective. Customers benefit from its always available support that proactively detects and solves potential issues, which is also a crucial feature for data-intensive enterprises. For more information, please visit [Actian.com](https://actian.com).

Questions we should ask

- What are the short- and long-term costs of the initiative?
- Does my development team have the expertise to do this project using hand coding?
- Is this how I want my hand-coding experts to spend their time?
- Can I do this same work with a tool cheaper and faster than my team can hand code?
- Is this a one-off, stand-alone project, or an area where I plan increasingly more development?
- How portable will this code be if I want to repurpose it on a new technology platform, such as
 - Spark or Flink?
- Will multiple developers collaborate on this project?
- How long will this code be in production?
- Who will own the maintenance of this code?
- How often will the code need to be updated?

“Data integration tools enable organizations to access, integrate, transform, process and move data spanning various endpoints and across any infrastructure to support their data integration use cases.”

-Gartner

Need more information?

Visit [Actian.com/dataconnect](https://actian.com/dataconnect) for product, service, solution information, papers, success stories, and more.