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

The Pros and Cons of Each Approach

"A homegrown IT approach may bring an initial 20 percent cost reduction, but result in a 200 percent increase in maintenance costs.[1]"

— Gartner (Source:[1] https://www.gartner.com/doc/3432617/does-customcoded-data-integration-stack)

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, yet, despite this lack of clarity, many companies still manage integration projects by writing custom code—typically to satisfy development efforts required 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 begin to mount.

Custom code can't be reused, it can be difficult 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 trade-offs 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 of these methods 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.

Need More information?

Visit www.actian.com/dataconnect for product, service, and solution information, papers, success stories and much more.

When considering initiating custom-coded projects, here's what IT decision makers must ask themselves:

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 projects. While your initial deployment costs might be reduced by 20% with a custom-coded approach, the maintenance costs will increase by 200%. If you want to build a repeatable and sustainable process on which your business can depend, then a data-integration tool may be a more sustainable choice.

If different people maintain and support the code once it's in production, then 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 the code leaves the company, then understanding how the integration job works (and how to fix it when it breaks) becomes exponentially harder 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 Hadoop or cloud platform, then who will do the work and how much ramp time will he or she 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 tool-based and custom-code solutions can only be delivered according to the ability and availability of their staff, meaning highly skilled staff with direct experience with the exact code and requirements.

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

Hand-coding experts are typically approximately a quarter of the full 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 doing so—, then wouldn't you rather have the experts doing something for which their unique skills are required?

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

Most IT teams are constantly being asked to do more with less. 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 is this an area where I plan to continue doing more and more future development?

There is a time and place for hand coding, but only in very specific situations. Custom coding can make sense for very targeted, simple and one-off projects that do not require much maintenance. It could also be necessary for situations in which there are no tools capable of doing the work required. If you are embarking on an initiative using a Big Data or cloud platform, then it's likely you will want to do more and more on that platform. If so, then relying on expert hand coders will be a very hard 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?

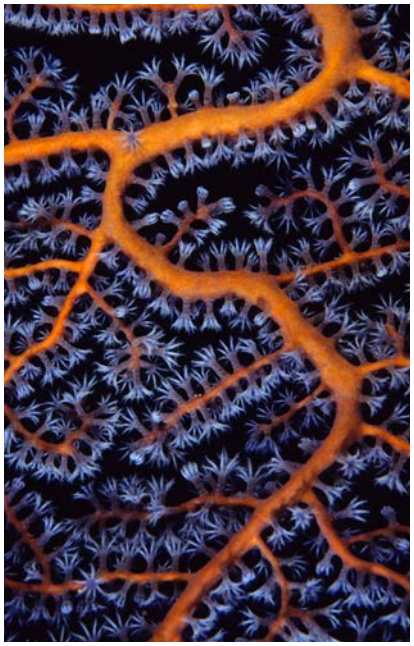
When budgeting your costs and time for an integration project, think about the additional efforts needed to re-develop all of your previous work in addition to the efforts needed for new development. Leading data-integration tools allow you to move simply from one data processing framework to another,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

7. Will multiple developers be collaborating on this project?

Understand that data-integration projects requiring multiple developers will benefit from the visual design environments provided by tools: easy reuse and code sharing, visual design environments, automated documentation and even wizards and experts to advise the developer. If you have several developers working on your integration initiative, then it is important to think about how your developers' work will fit together. With custom coding, there is no guaranteed consistency from one developer to the next, which can make development and maintenance complex and costly. A tool with the option to reuse prior development elements will keep your integration team from duplicating effort and result in more efficient data integration flows.

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

10. How often will the code need to be updated to accommodate new business needs or changes in the 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. a Hybrid Integration Tool

Category	Hand Coding	Hybrid Integration Solution
Maintenance Costs	**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 audit 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 deliver an optimized solution without the pain of iterative development.
Changes to Business Processes & APIs	**Very expensive and time consuming** to manage, requiring skilled programmers.	**Dynamic configuration** changes to cope with business process changes and APIs.
ROI	**Negative.** 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 Hybrid 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.

www.actian.com | **Toll Free** +1 888 446 4737 | **Tel** +1 650 587 5500
2300 Geng Rd., Suite 150, Palo Alto, CA 94303