When should we use ETL tools rather than Stored Procedures for ETL process?

Posted on [April 25, 2014](https://msbigurukool.wordpress.com/2014/04/25/when-should-we-use-etl-tools-rather-than-stored-procedures-for-etl-process/)

If your ETL is mostly E (Extract) and L (Load), with very little of T (Transformation) and your SPs don’t rely on cursors, then one can go with Stored Procedures.

If your ETL is mostly E (Extraction) and L (Loading), with very little T (Transformation), and if you can write your SPs so they don’t rely on cursors, then going the SP-route only is probably fine.

But for more complex processes, particularly those that involve heavy transforms, slowly changing dimensions, data mining lookups, etc, an ETL tool has some advantages:

Advantages of ETL:

* **Visual flow and self-documentation:** The single greatest advantage of an ETL tool is that it provides a visual flow of system’s logic. Each ETL tool presents these flows differently, but even the least-appealing of the UIs compare favorable to customary stored procedures.
* **Structured system design:** ETL tools are designed for specific problem of populating a data warehouse. Although they are mere tools, they do provide a metadata-driven structure to the development team. This is particularly valuable for teams building their first ETL system.
* **Operational resilience:**Many of the home-grown ETL have too many operational problems. ETL tools provide functionality and practices for operating and monitoring the ETL system in production. You can certainly design and build a well instrumented hand-coded ETL application, and ETL tool operational features have yet to mature. Nonetheless, it’s easier for a data warehouse / business intelligence team to build on the management features of an ETL tool to build a resilient ETL system.
* **Data-lineage and data-dependency functionality.** We would like to be able to right-click on a number in a report and see exactly how it was calculated, where the data was stored in the data warehouse, how it was transformed, when the data was most recently refreshed, and what source system or systems underlay the numbers. *Dependency* is the flip side of lineage: we’d like to look at a table or column in the source system and know which ETL modules, data warehouse tables, OLAP cubes, and user reports might be affected by a structural change. In the absence of ETL standards that hand-coded systems could conform to, we must rely on ETL tool vendors to supply this functionality — though, unfortunately, few have done so to date.
* **Advanced data cleansing functionality.** Most ETL systems are structurally complex, with many sources and targets. At the same time, requirements for transformation are often fairly simple, consisting primarily of lookups and substitutions. If you have a complex transformation requirement, for example if you need to de-duplicate your customer list, you should use a specialized tool. Most ETL tools either offer advanced cleansing and de-duplication modules (usually for a substantial additional price) or they integrate smoothly with other specialized tools. At the very least, ETL tools provide a richer set of cleansing functions than are available in SQL.
* **Performance.** You might be surprised that performance is listed last under the advantages of the ETL tools. It’s possible to build a high-performance ETL system whether you use a tool or not. It’s also possible to build an absolute dog of an ETL system whether you use a tool or not. I’ve never been able to test whether an excellent hand-coded ETL system outperforms an excellent tool-based ETL system; I believe the answer is that it’s situational. But the structure imposed by an ETL tool makes it easier for an inexperienced ETL developer to build a quality system.

1. It manages memory very efficiently, which can result in big performance improvements compared to Stored Procedure
2. The graphical interface lets you build large, complex and reliable transforms much more easily than hand-crafted stored procedures.
3. ETL tools lets you more easily interact with additional external resources, which can be very handy for things like data cleansing.

**TO SUM IT UP**

Should you use an ETL tool? Yes

Do you have to use an ETL tool? No

For teams building their first or second ETL system, the main advantage of visual tools are self-documentation and a structured development path. For newbies, these advantages are worth the cost of the tool. If you’re a seasoned expert – perhaps a consultant who has built dozens of ETL systems by hand – it’s tempting to stick to what has worked well in the past this level of expertise, you can probably build a system that performs as well, operates as smoothly, and perhaps costs less to develop than a tool-based ETL system. But many seasoned experts are consultants, so you should think objectively about how maintainable and extensible a hand-crafted ETL system might be once consultant has moved on.