

Summary of Modeling

The process of building a new application can be divided into two primary aspects:

- Data modeling (schema)
- Interaction (business rules, UX)

This is done on an existing system with standard products from Aras, Modules from 3rd Parties, or earlier work by the user's company for various reasons.

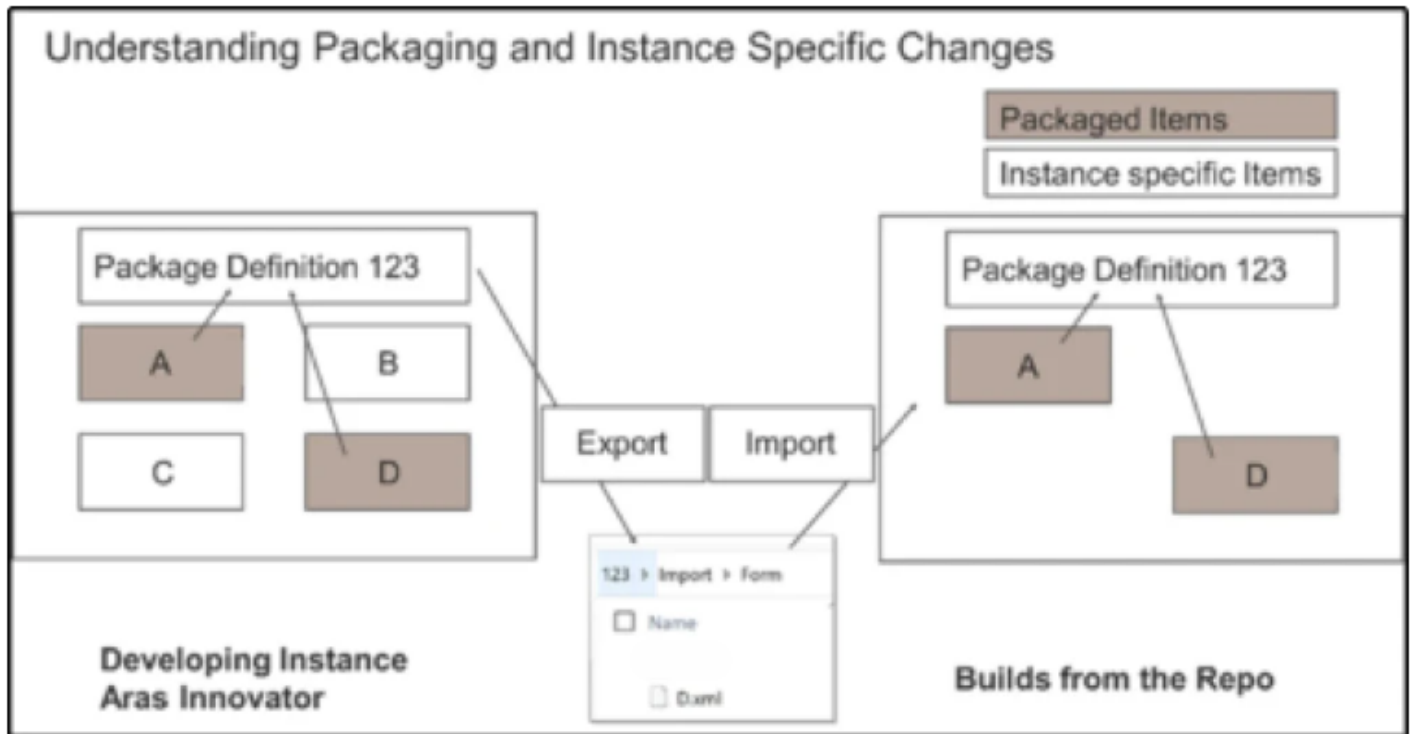
The new applications must consider all the existing work when modularizing, packaging, and building.

The diagrams below illustrate that the new application can be simple or ambitious.



The diagram below summarizes the impact of anonymous items in an instance and the effect of defining, exporting, and providing **Packages** to the Build System.





On the left, the user has four items (**A**, **B**, **C**, and **D**). **D** represents customer-created or modified items properly packaged, exported, copied, and assigned to the change control system Git. **A** represents Core Released AML Item these are always present in every Innovator instance, but are not stored in the repository. If a core item has been customized, only the delta the specific change is stored in the repository rather than the full item. **B** and **C** represent Innovator instance-specific items not intended for use in the next Build and, therefore, are not packaged.

The Build System then produces the instance on the right. It's important to observe that the Aras Innovator instance on the right excludes items **B** and **C**, showcasing the capability to dictate what DevOps builds. Core Released AML Item (**A**) is always included in the instance by the platform, regardless of whether they appear in the repository. This capacity to specify what DevOps creates is a foundational element of utilizing DevOps.