Baseline Management

When a project team receives the SDE from Aras, a **Baseline** is established, which acts as the starting point for all future changes to the software. The execution of the setup script (**BuildAndDeploy.ps1**) installs an initial database and sets up all required files in the code tree directory.

The established **Baseline** becomes the foundation for layering changes each time the setup scripts are run.

As these changes accumulate over time, they may grow significantly larger. Therefore, when the solution enters production, a fresh **Baseline** incorporating all customizations can be set. This new **Baseline** will be the starting point for the setup scripts.

After Aras has provided an initial **Baseline**, the project team can make modifications as needed. This may include:

- **Add new applications to the platform**: Adding applications to a platform involves enhancing its functionality and broadening the system's capabilities. The Adding Applications to a Project Appendix demonstrates an example of integrating an application into Aras Innovator.
- **Add Language packs**: Adding language packs to software is crucial in making applications accessible and user-friendly to a diverse global audience.
- **Establish new Baselines**: A new **Baseline** provides a snapshot of the project's status, including what has been achieved and the resources expended to reach this point. Once established, this new **Baseline** serves as the starting point for subsequent phases or steps in the project. For more information, refer to the Generate New Baseline section.
- **Deploy Build to SIT (for QA)**: SIT involves testing the system as a whole in an environment that closely mirrors production to ensure that all integrated components work together as expected. This includes ensuring new applications function correctly with the existing system and language packs work as intended. For more details, refer to the Deploy to System Integration Testing (SIT) Environment section.

When an SDE from Aras is received and a **Baseline** is established, it is not a final process. Instead, it is an ongoing effort, and modifications to the software are carried out over time as the project

©2026 Aras Corporation All Copyrights Reserved

requirements evolve. These modifications might include integrating new applications, adding new features, fixing bugs, and improving system performance, among other things.

However, this process should be handled appropriately. Software changes must align with the project's goals and should not introduce new issues or conflicts. Therefore, comprehensive testing should be performed after each modification to verify that the changes are working as expected.

Reducing build time is another important aspect of this process. When modifications are organized and managed properly, the time required to build the software can be significantly reduced. This efficiency can lead to quicker deployments and a shorter overall time to market, which can be a significant advantage for the project.

Finally, all these activities must be done as part of a deployment policy. A deployment policy outlines the standards and procedures for making changes to the software, testing those changes, and deploying the software to the production environment. This policy helps ensure that all changes are carried out in a controlled and consistent manner, which can contribute to the overall quality and success of the project.

©2026 Aras Corporation All Copyrights Reserved