

# Partial Local Deployment

If you already have a deployed instance of Aras Innovator with data (see section 3.2.6 Build and Deploy Locally), and you want to iteratively update it during development, you can perform a partial deployment to speed up the process and avoid unnecessary steps.

By default, `.\BuildAndDeploy.ps1` execution without any parameters will perform the following actions:

1. Validate and install build machine prerequisites
2. Remove installed Aras Innovator instance
3. Install new Aras Innovator instance from baseline
4. Build `CustomSolutions.sln`
5. Build new deployment package with all kinds of customizations such as AML changes, Code Tree changes, and Configuration transformation scripts
6. Deploy the package on top of the new Aras Innovator instance.

As a result, each time the `BuildAndDeploy.ps1` script is executed without parameters, a fresh instance is deployed from scratch — even for minor changes like updating the login page image.

To accelerate development and avoid re-running unnecessary steps, you can use delta deployment option parameters of `BuildAndDeploy.ps1` for partial deployments.

## Important

A full deployment must be executed at least once before using any delta deployment options. These options update only an existing instance and will fail if no instance is found.

## Important

If `.\BuildAndDeploy.ps1` with delta deployment options failed or you are not sure that update can be deployed on top of existing Aras Innovator instance, please use `.\BuildAndDeploy.ps1` without any parameters.

## Deployment Option Parameters

### -UpdateInstance

Deploys all types of customizations (AML changes, CodeTree changes, and Configuration transformation scripts) on top of the existing Aras Innovator instance without removing it.



```
BuildAndDeploy.ps1 -UpdateInstance  
-UpdateAMLOnly
```

Deploys only AML customizations to the database of the existing Aras Innovator instance.

```
BuildAndDeploy.ps1 -UpdateAMLOnly  
-UpdateCodeTreeOnly
```

Deploys only the CodeTree customizations (e.g., server-side and client-side code changes) without affecting AML or configuration files.

```
BuildAndDeploy.ps1 -UpdateCodeTreeOnly  
-UpdateConfigurationOnly
```

Deploys only configuration transformation customizations (e.g., changes in config files in TransformationsOfConfigFiles folder) on top of the existing instance.

```
BuildAndDeploy.ps1 -UpdateConfigurationOnly
```

#### **Important**

Deployment package that is built during local delta deployment is stored in AutomatedProceduresOutput/UpdatePackage. Do not use this deployment package outside of the existing instance as far as it is partial deployment package and can't be reused for other deployments. To determine what kind of customizations are in the update deployment package, you can look at "PackageInfo.md" from the package content.

#### **Skip Prerequisite Installation**

When running a full `.\BuildAndDeploy.ps1`, the script validates and installs build machine prerequisites and restores packages. This step is also executed by default when running delta deployments, which can take time even if everything is already set up.

If you are sure that the Aras.DevOps packages are already restored and prerequisites are installed, you can skip this step using the `-SkipPrerequisites` parameter:

```
BuildAndDeploy.ps1 -SkipPrerequisites
```

#### **Important**



Do not skip prerequisites when updating to a new version of the Aras.DevOps packages. Run the full BuildAndDeploy.ps1 script first to ensure everything is validated and restored properly. After that, you may add -SkipPrerequisites to reduce execution time for subsequent deployments.

#### **Example: Fastest Way to Deploy Code Customization**

If you have an already configured environment and an existing deployed instance, and you only want to update the CodeTree (e.g., UI or server code), use the following command for the fastest deployment:

```
BuildAndDeploy.ps1 -UpdateCodeTreeOnly -SkipPrerequisites
```

This will apply CodeTree customizations directly to the existing instance without re-installing or validating prerequisites.

