

Local Environment Variables Set Up

A local environment is now established and connected to the Standard SDE.

The build and deployment scripts running in the SDE possess local equivalents that require specific details related to the given environment and the associated Git **Branch**. Property settings provide these details.

The property values have the following precedence. Based on this precedence, they can be set or overridden. The highest is evaluated last, as shown.

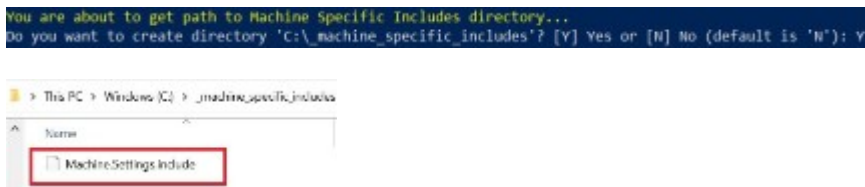
- **Default.Settings.include:** This file is located in ...\\AutomatedProcedures\\
- **Machine.Settings.include:** This file is located in C:\\ or an alternate root directory.
- **<ProjectPrefix>-<git-branch>.Settings.include:** The project Prefix is set in Default.Settings.include the Git **Branch** must match the **Branch** the user checks out.

The BuildAndDeploy.ps1 file is used to deploy Aras Innovator and the customization from the current directory.

The following steps outline the process to run the **BuildAndDeploy.ps1** script in a local environment:

1. Open Windows PowerShell as administrator and navigate to the working directory. For example, C:\\ArasProjects\\project1\\localRepo
2. Run the command: **.\BuildAndDeploy.ps1**

When the **BuildAndDeploy.ps1** script is run for the first time, it does several environment verifications and creates the following file: C:_machine_specific_includes\\Machine.Settings.include



The Machine.Settings.include file contains a series of key/value pairs that are used by the build process to build Aras Innovator on the local machine.



The project prefix and a reference to the last **Commit** are used from an Include file in the following Local Repository: C:\ArasProjects\project1\localRepo\AutomatedProcedures\Default.Settings.include

The Default.Settings.include includes all the default properties required for a project.

When executing a script, a Variable is first loaded from Default.Settings.include. If the same Variable is defined in the Machine.Setting.include, the value is overwritten. Finally, the value is overwritten if the Variable is defined in a **Branch**-specific file.

3. Update the Machine.Setting.include file with the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<project name="Default.Settings">
  <property name="Path.To.Baselines.Dir" overwrite="true" value="C:\ArasProjects\Baselines" />
  <property name="Path.To.DB.Bak" overwrite="true" value="C:\ArasProjects\Baselines\ArasSSETraining\CleanInnovator14SP3\DB.bak" />
  <property name="Path.To.CodeTree.Zip" overwrite="true" value="C:\ArasProjects\Baselines\ArasSSETraining\CleanInnovator14sp3\CodeTree.zip" />
  <property name="MSGQ.Server" overwrite="true" value="" />
  <property name="MSGQ.SA.Password" overwrite="true" value="" />
  <property name="MSGQ.Innovator.Password" overwrite="true" value="" />
  <property name="MSGQ.Innovator.Regular.Password" overwrite="true" value="" />

  <!-- Note: If you want to set licenses here, please make sure they have been removed from 'BranchSpecificSettingsFile' to avoid overwriting. -->
  <choose>
    <when test="$([string]:starts-with((Version.Of.Installed.Innovator, 11))" >
      <!-- Here you can set all the properties specific to Innovator 11, in particular, the licenses -->
      <property name="Innovator.License.Type" overwrite="true" value="" />
      <property name="Innovator.License.Company" overwrite="true" value="" />
      <property name="Innovator.License.Key" overwrite="true" value="" />
      <property name="Innovator.Activation.Key" overwrite="true" value="" />
      <property name="Feature.License.Strings.List" overwrite="true" value="" />
    </when>
    <when test="$([string]:starts-with((Version.Of.Installed.Innovator, 12))" >
      <!-- Here you can set all the properties specific to Innovator 12, in particular, the licenses -->
      <property name="Innovator.License.Type" overwrite="true" value="" />
      <property name="Innovator.License.Company" overwrite="true" value="" />
      <property name="Innovator.License.Key" overwrite="true" value="" />
      <property name="Innovator.Activation.Key" overwrite="true" value="" />
      <property name="Feature.License.Strings.List" overwrite="true" value="" />
    </when>
    <when test="$([string]:starts-with((Version.Of.Installed.Innovator, 14))" >
      <!-- Here you can set all the properties specific to Innovator 14, in particular, the licenses -->
      <property name="Innovator.License.Type" overwrite="true" value="Unlimited" />
      <property name="Innovator.License.Company" overwrite="true" value="Aras Corporation" />
      <property name="Innovator.License.Key" overwrite="true" value="" />
      <property name="Innovator.Activation.Key" overwrite="true" value="" />
      <property name="Feature.License.Strings.List" overwrite="true" value="" />
    </when>
  </choose>
</project>
```

Project-Specific Settings:

- **Path to baseline directory:** For example, C:\ArasProjects\Baselines
- **Path.To.DB.Bak:** File path to a “clean” copy of the Aras Innovator solutions database. For example, C:\ArasProjects\project1\Baselines\CleanInnovatorxxSPyy\DB.bak
- **Path.To.CodeTree.Zip:** A file path to the CodeTree.zip archive containing the code tree of the production Aras Innovator instance. For example, C:\ArasProjects\project1\Baselines\CleanInnovatorxxSPyy\CodeTree.zip

Machine-Specific Properties:

- **Innovator.License.Type:** This is typically Unlimited, Version, or Verified, depending on the key being used for this project.
- **Innovator.License.Company:** The licensed company name.



- **Innovator.License.Key:** A license key used for the installation can be obtained from <http://www.aras.com/support/LicenseKeyService/> .
- **Innovator.Activation.Key:** An activation key for features that have been licensed for this project.
- **MSSQL.Server:** Name of the SQL Server instance (default is the local machine).
- **MSSQL.SA.Password:** The password for the sa (system administrator) login.
- **MSSQL.Innovator.Password:** Password for the Aras admin login (default is innovator).
- **Feature.License.Strings.List:** This is an optional property. It is required only when the user needs to set values for features such as TAF (Test Automation Framework) which is required as part of CI/CD.

Important

In SaaS, ensure that MSSQL.Innovator.Password and MSSQL.Innovator.Regular.Password meet Microsoft SQL Server complexity requirements. Passwords must comply with [SQL Server Password Policy](#) to be used with contained databases.

Failure to use complex passwords will result in BuildAndDeploy.ps1 errors during database setup.

