



Aras Test Automation (TAF) Framework 1.45 Upgrade Guide

Document #: D-008085

Last Modified: 6/5/2024

Copyright Information

Copyright © 2024 Aras Corporation. All Rights Reserved.

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Phone: 978-691-8900

Notice of Rights

Copyright © 2024 by Aras Corporation and/or its affiliates. All rights reserved.

This document is protected by U.S. and international copyright laws and conventions. No copyright may be obscured or removed from this document. This document may not be modified or altered, or reproduced or transmitted in any form, without the explicit permission of the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Notice of Liability

THIS DOCUMENT IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY, AND THE CONTENTS HEREOF ARE SUBJECT TO CHANGE WITHOUT NOTICE. THE INFORMATION CONTAINED IN THIS DOCUMENT IS DISTRIBUTED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR A WARRANTY OF NON-INFRINGEMENT. ARAS SHALL HAVE NO LIABILITY TO ANY PERSON OR ENTITY WITH RESPECT TO ANY LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY THE INFORMATION CONTAINED IN THIS DOCUMENT OR BY THE SOFTWARE OR HARDWARE PRODUCTS DESCRIBED HEREIN.

Table of Contents

Send Us Your Comments	4
1 Overview	5
2 Prerequisites.....	5
3 Upgrade Process.....	5
3.1 Updating the Templates for New Projects	5
3.1.1 <i>Uninstalling a previous TAF version from Visual Studio 2019</i>	<i>5</i>
3.1.2 <i>Installing a new Aras TAF version in Visual Studio 2019.....</i>	<i>7</i>
3.1.3 <i>Uninstalling a Previous TAF version from Visual Studio 2022.....</i>	<i>9</i>
3.1.4 <i>Installing a new TAF version in Visual Studio 2022</i>	<i>10</i>
3.2 Updating Existing Projects	11
3.2.1 <i>Updating NUPKG files.....</i>	<i>11</i>
3.2.2 <i>Updating an integration test created with TAF 1.11 or Earlier</i>	<i>12</i>
3.2.3 <i>Updating a Selenium test created with TAF 1.20 or Earlier</i>	<i>13</i>
3.2.4 <i>Updating a Selenium test created with TAF 1.29 or Earlier for Aras Innovator 14 or Later</i>	<i>16</i>
3.2.5 <i>Updating Existing Solution created with TAF 1.29 or earlier for Aras Innovator 14 or later from External Innovator versioning approach to Internal one.....</i>	<i>17</i>

Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

Email:

TechDocs@aras.com

Subject: Aras Product Documentation

Or,

Postal service:

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Attention: Aras Technical Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit <https://www.aras.com/support/>.

1 Overview

This document describes how to:

- Upgrade the Aras Test Automation Framework from the previous version to 1.45.
- Update templates for new projects.
- Update existing projects.

2 Prerequisites

To upgrade the Aras Test Automation Framework to version 1.45, users must have any of its previous versions (1.0 – 1.44) installed.

For the software and license requirements, see the *Aras Test Automation Framework 1.45 – Installation Guide*.

3 Upgrade Process

The upgrade process scenario is updating Aras TAF items in the following order:

1. Aras TAF project templates.
2. Existing projects.

3.1 Updating the Templates for New Projects

To update the Visual Studio extension with the Aras TAF project templates for AML integration and Selenium web UI tests:

1. Uninstall the previous Aras Test Automation Framework version.
2. Install a new Aras Test Automation Framework version.

3.1.1 Uninstalling a previous TAF version from Visual Studio 2019

1. Run Visual Studio 2019.
2. In the main menu, go to **Extensions --> Manage Extensions**.

3. In the appeared **Manage Extensions** dialog, select **Aras.TAF.Templates** and click **Uninstall**.

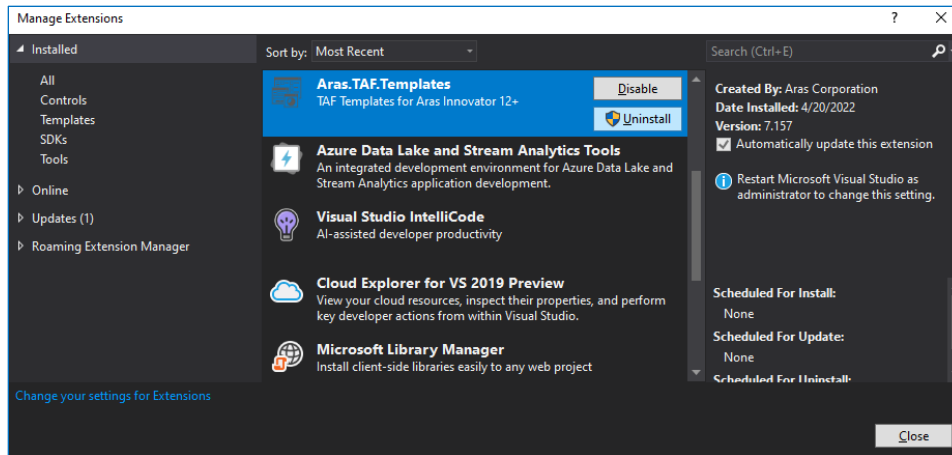


Figure 1. **Aras.TAF.Templates** in the **Manage Extensions** dialog

4. In the appeared **Microsoft Visual Studio** confirmation dialog, click **Yes**.

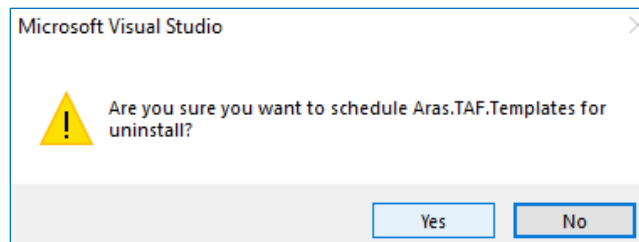


Figure 2. The **Microsoft Visual Studio** confirmation dialog

5. Close the **Manage Extensions** dialog and Visual Studio. The **User Account Control** dialog appears.

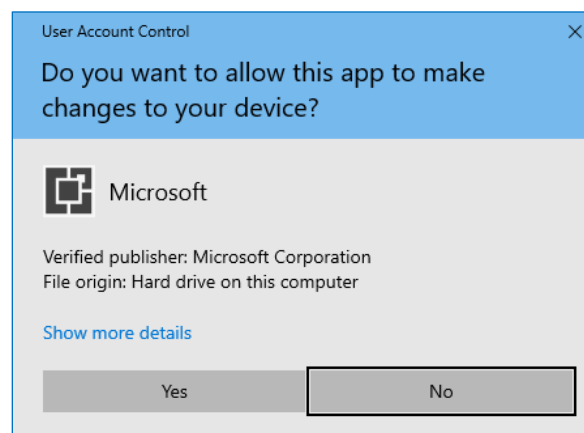


Figure 3. The **User Account Control** dialog

6. In the **User Account Control** dialog, click **Yes**. The **VSIX Installer** dialog appears.

7. In the **VSIX Installer** dialog, click **Modify**. The uninstallation process begins.

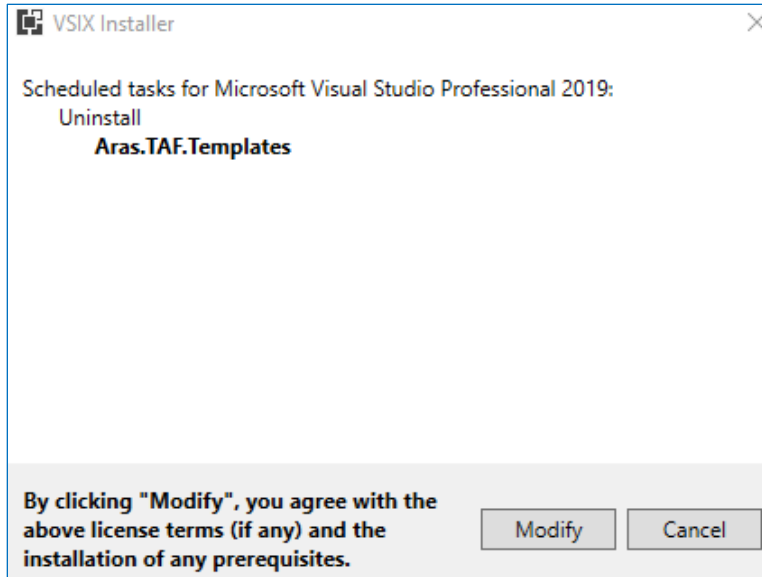


Figure 4. The **VSIX Installer** dialog asking to start the plugin uninstallation

8. Once the uninstallation is successfully completed, close the **VSIX Installer** dialog.

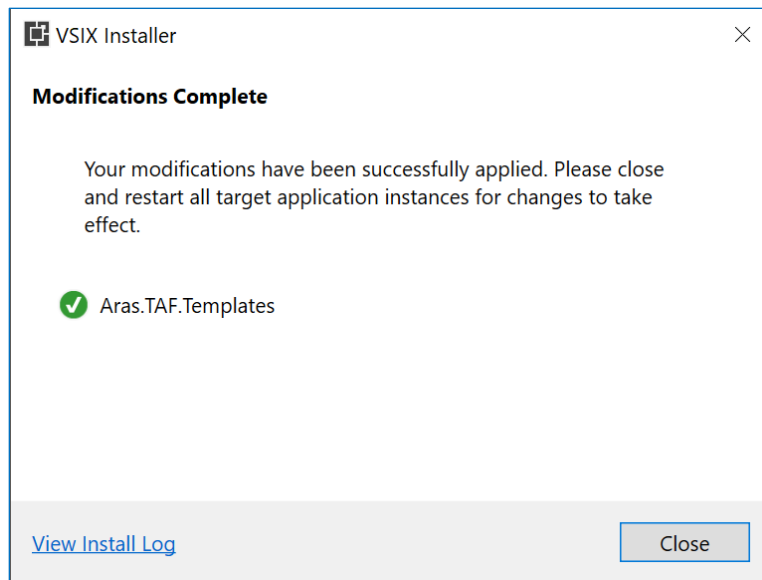


Figure 5. The **VSIX Installer** dialog notifying about the successful plugin uninstallation

3.1.2 Installing a new Aras TAF version in Visual Studio 2019

1. Unzip the Aras TAF package on a machine with installed Visual Studio 2019.
2. Go to the **Templates** folder in the unzipped package.
3. Double-click the **Aras.TAF.Templates.Installer.vsix** file to start the installer. The **User Account Control** dialog pops up.

- Click **Yes**. The **VSIX Installer** dialog pops up.

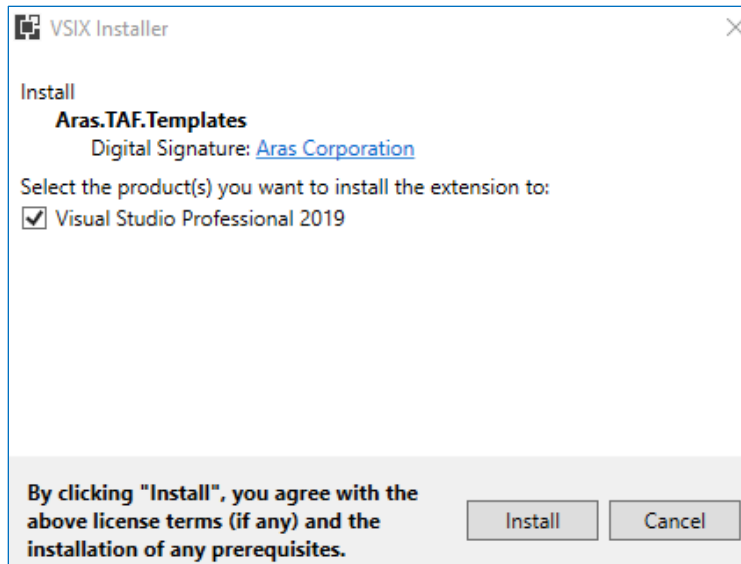


Figure 6. The **VSIX Installer** dialog asking to choose a Visual Studio version

- Ensure that the Visual Studio 2019 option is selected.
- Click the **Install** button. The installation process begins.

Note: If you have open Visual Studio processes, the installer will ask you to end them. If so, save the open Visual Studio projects, and click the **End Tasks** button to stop the processes and continue the installation.

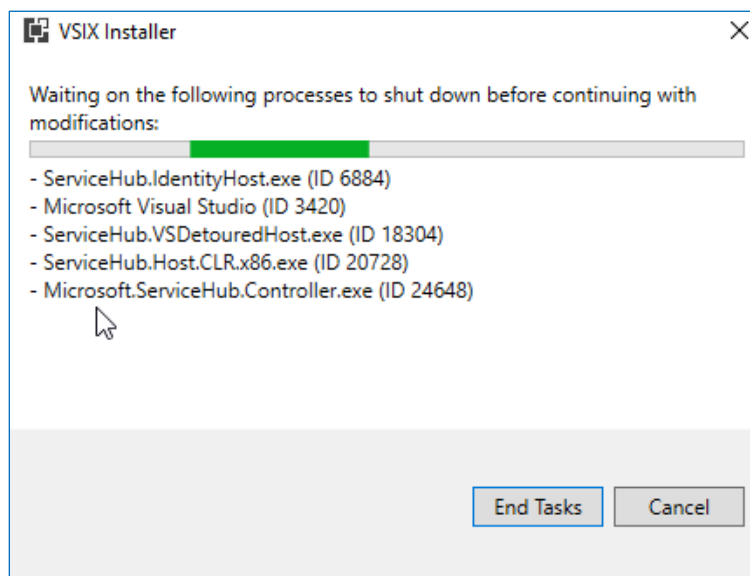


Figure 7. The **VSIX Installer** dialog asking to end all running Visual Studio processes

Once the extension with TAF templated is installed, the dialog informs about the successful installation.

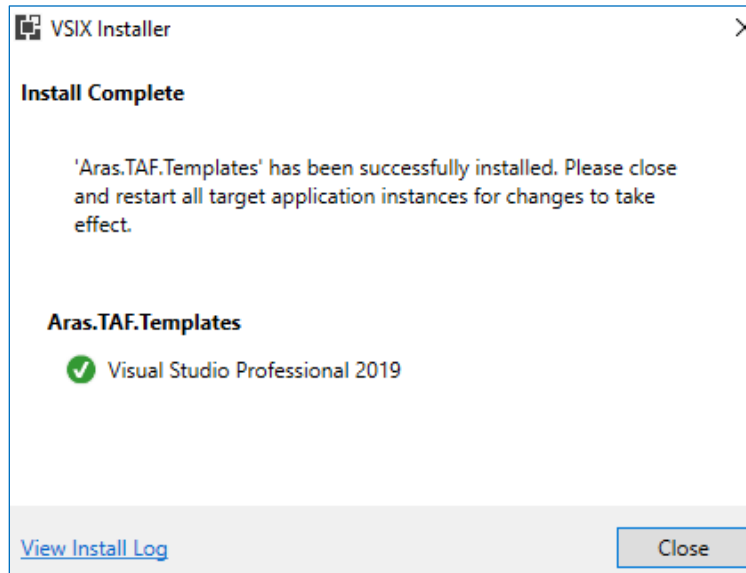


Figure 8. The **VSIX Installer** dialog informing of a successful installation

7. Click the **Close** button.

3.1.3 Uninstalling a Previous TAF version from Visual Studio 2022

1. Open PowerShell console and navigate to the Templates folder in the unzipped package.
2. Fill `./UninstallTemplate.ps1` to the PowerShell console.

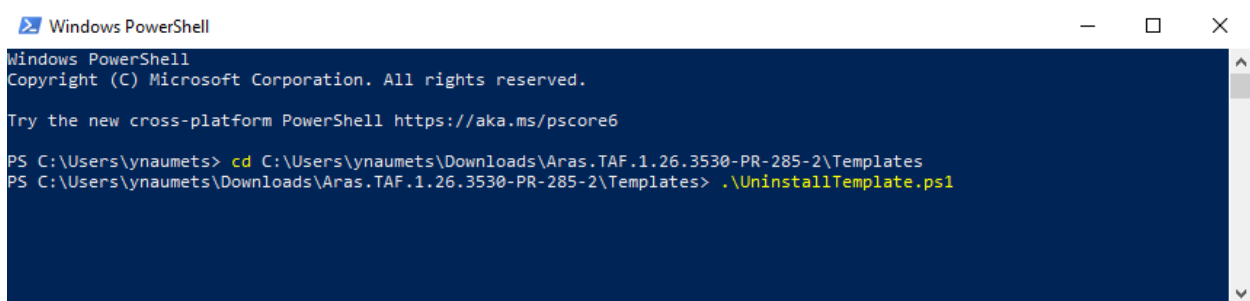


Figure 9. The PowerShell console with Uninstallation command.

3. Click **Enter**.

```

Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ynaumets> cd 'C:\Users\ynaumets\Downloads\Aras.TAF.1.26.3572\Templates'
PS C:\Users\ynaumets\Downloads\Aras.TAF.1.26.3572\Templates> .\UninstallTemplate.ps1
Uninstalling Aras TAF Templates...
Warning: use of 'dotnet new --uninstall' is deprecated. Use 'dotnet new uninstall' instead.
For more information, run:
    dotnet new uninstall -h

Success: Aras.TAF.Templates::1.26.3572 was uninstalled.
PS C:\Users\ynaumets\Downloads\Aras.TAF.1.26.3572\Templates>
    
```

Figure 10. The Powershell console notifying about the successful plugin uninstallation

3.1.4 Installing a new TAF version in Visual Studio 2022

1. Unzip the Aras TAF package on a machine with installed Visual Studio 2022.
2. Go to the Templates folder in the unzipped package.
3. Open PowerShell console and navigate to the Templates folder in the unzipped package.
4. Fill `.\InstallTemplate.ps1` to the PowerShell console

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jjacobellis> cd C:\Users\jjacobellis\Downloads\4162\Aras.TAF.1.45.4162\Templates
PS C:\Users\jjacobellis\Downloads\4162\Aras.TAF.1.45.4162\Templates> .\InstallTemplate
    
```

Figure 11. The Powershell console with Installation command

5. Click **Enter**.

Once the extension with TAF templates is installed, the PowerShell console informs about the successful installation.

```

Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jjacobellis> cd C:\Users\jjacobellis\Downloads\4162\Aras.TAF.1.45.4162\Templates
PS C:\Users\jjacobellis\Downloads\4162\Aras.TAF.1.45.4162\Templates> .\InstallTemplate
Installing Aras TAF Templates...
The following template packages will be installed:
    C:\Users\jjacobellis\Downloads\4162\Aras.TAF.1.45.4162\Templates\Aras.TAF.Templates.1.45.4162.nupkg

Success: Aras.TAF.Templates::1.45.4162 installed the following templates:
Template Name                Short Name    Language    Tags
-----
Aras Innovator 12 Selenium Tests Template  I12Template  [C#]       Test/NUnit/Selenium
Aras Innovator 14+ Selenium Tests Template  I14+Template [C#]       Test/NUnit/Selenium
Aras Integration Tests Template           ItafTemplate [C#]       Test/NUnit/Integration
    
```

Figure 12. The Powershell console informs about a successful installation.

3.2 Updating Existing Projects

To update Visual Studio projects created with an older Aras Test Automation Framework version:

1. Update the NUPKG files as discussed in subsection [3.2.1](#).
2. If you are updating:
 - An integration test created with TAF 1.11 or earlier, do the procedure discussed in subsection [3.2.2](#).
 - A Selenium test created with TAF 1.20 or earlier, do the procedure discussed in subsection [3.2.3](#).
 - Either:
 - An integration test created with TAF 1.12 or later.
 - A Selenium test created with TAF 1.21 or later.
3. Rebuild your project.

3.2.1 Updating NUPKG files

Note: We strongly recommend backing up the NUPKG files to enable reverting to the pre-upgrade state of the project.

1. Go to the **libraries\TAF** folder in the project repository and copy all NUPKG files to a backup location.

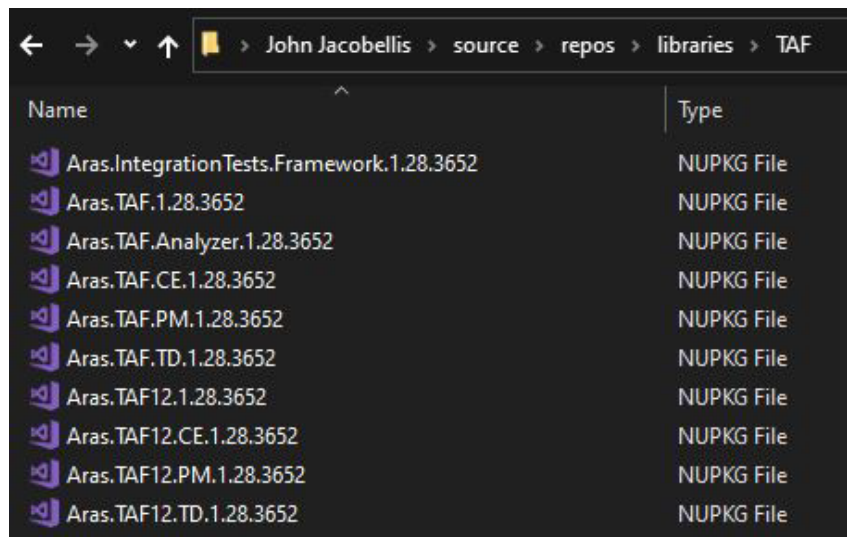


Figure 13. TAF library files in the **libraries\TAF** project repository folder

2. After the backup is completed, delete all NUPKG files from the **TAF** project repository folder.
3. Copy the unzipped Aras Test Automation Framework 1.38 package to the given machine.

- Copy all NUPKG files from the **Libraries** package folder to the **TAF** project folder.

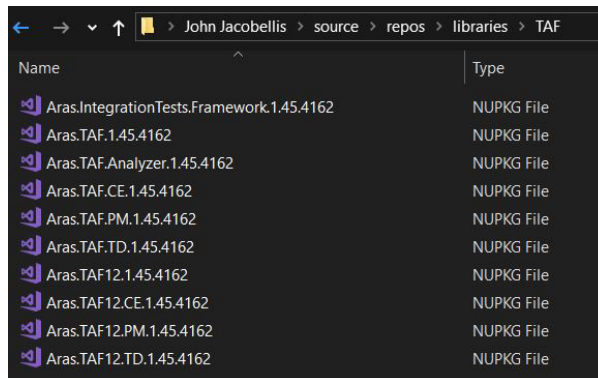


Figure 14. TAF library files in the **Libraries** package folder

3.2.2 Updating an integration test created with TAF 1.11 or Earlier

Note: Without this update, an integration test created with TAF 1.11 or earlier will fail during execution time.

- Open the project in Visual Studio.
- Right-click the project in the Solution Explorer and click **Edit Project File**.

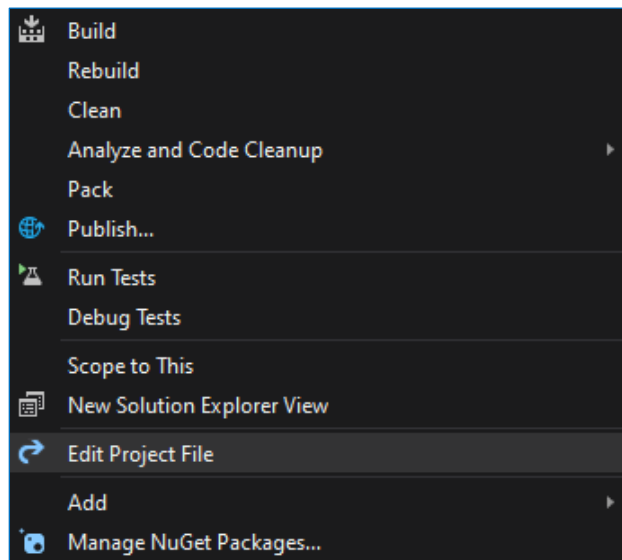


Figure 15. The **Edit Project File** command

- In the `PropertyGroup` tag of the **Aras.ITAF.Tests.csproj** file, add the `GenerateBindingRedirectsOutputType` child tag with the `true` value as shown in the example below:

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net472</TargetFramework>
    <Deterministic>true</Deterministic>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
</Project>
```

```

    <GenerateBindingRedirectsOutputType>true</GenerateBindingRedirect
sOutputType>
  </PropertyGroup>
  ...

```

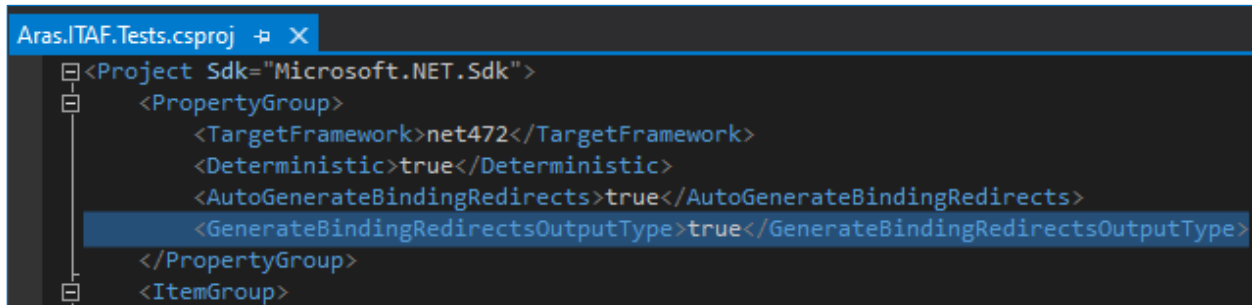


Figure 16. The GenerateBindingRedirectsOutputType tag

4. Save the **Aras.ITAF.Tests.csproj** file.
5. Rebuild the solution.

After the project is rebuilt, the new version of the TAF libraries should be in the Solution Explorer.

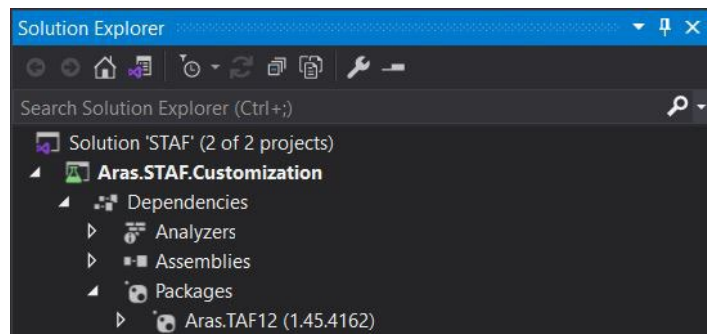


Figure 17. A rebuilt ITAF project with the updated TAF libraries

3.2.3 Updating a Selenium test created with TAF 1.20 or Earlier

Note: Without this update, a Selenium test created with TAF 1.20 or earlier will be executed, but the following MSB3277 warning will be raised: Found conflicts between different versions of "System.ValueTuple" that could not be resolved.

1. Right-click the project in the Solution Explorer and click **Edit Project File**.
2. In the `PropertyGroup` tag of the **Aras.STAF.Tests.csproj** file, add the `AutoGenerateBindingRedirects` and `GenerateBindingRedirectsOutputType` child tags with the `true` values as shown in the example below:

```

<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
  ...
  <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  <GenerateBindingRedirectsOutputType>true</GenerateBindingRedirectsOu
tputType>
  </PropertyGroup>

```

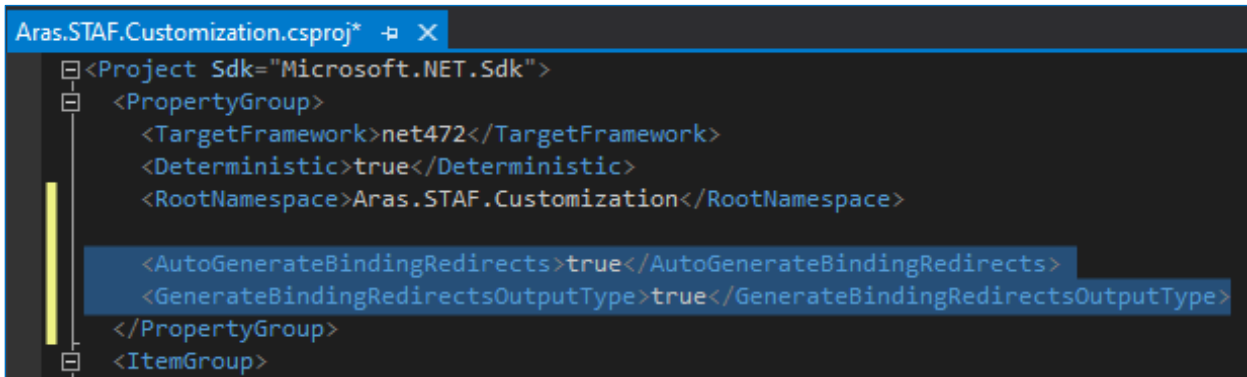


Figure 18. The tags to be added to the **Aras.STAF.Tests.csproj** file

3. Save the **Aras.STAF.Tests.csproj** file.
4. In the Visual Studio Solution Explorer, double-click the **app.config** file to open it.

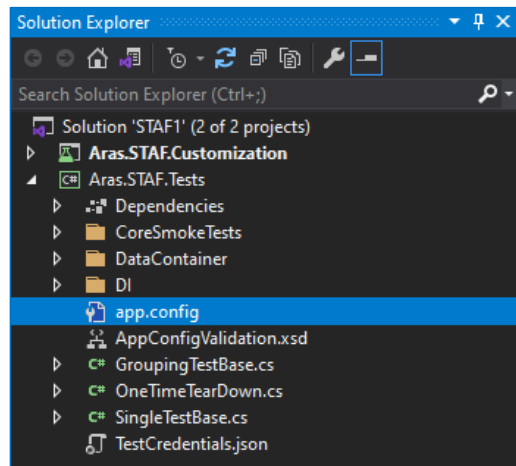


Figure 19. The **app.config** file

5. In the **app.config** file, remove the following `<dependentAssembly>` tag:

```
<configuration>
...
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="System.ValueTuple"
publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.0.3.0"
newVersion="4.0.3.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>...</configSections>
  <MyArasSettings xsi:noNamespaceSchemaLocation="AppConfigValida"...>...</MyArasSettings>

  <startup>...</startup>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="System.ValueTuple" publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.0.3.0" newVersion="4.0.3.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Ninject" publicKeyToken="c7192dc5380945e7" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-3.3.4.0" newVersion="3.3.4.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>

```

Figure 20. The dependentAssembly tag to be removed

6. Save the **app.config** file.
7. Rebuild the solution.

After the project is rebuilt, the new version of the TAF libraries should be in the Solution Explorer.

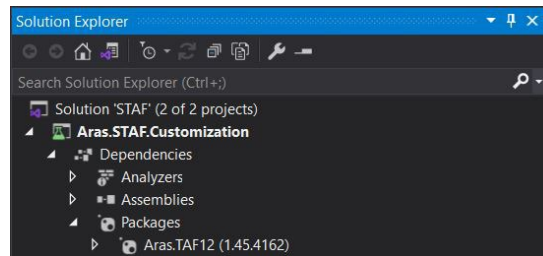


Figure 21. A rebuilt STAF project with the updated TAF libraries

3.2.4 Updating a Selenium test created with TAF 1.29 or Earlier for Aras Innovator 14 or Later

Note: Without this update, the “S_1_002_LoginProcedure” Selenium test created with TAF 1.29 or earlier for Aras Innovator 14 or higher will fail during execution time.

1. Install the TAF 1.32 Template (NuGet for VS2022 or VSIX for VS2019) – see section 3.1 of Upgrade Guide.
2. Create a new solution using the **Aras Innovator 14+ Selenium Tests Template** – see section 7.1 of Installation Guide.
3. Replace the “S_1_002_LoginProcedure.cs” file of your existing solution with the “S_1_002_LoginProcedure.cs” file from the newly created solution.

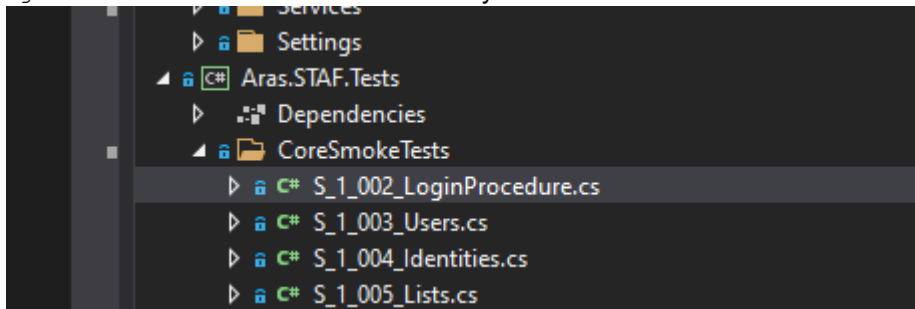


Figure 22. S_1_002_LoginProcedure.cs

4. Replace the “S_1_002_ja_i26.json” file of your existing solution with the “S_1_002_ja_i14_0_12.json” file from the newly created solution.

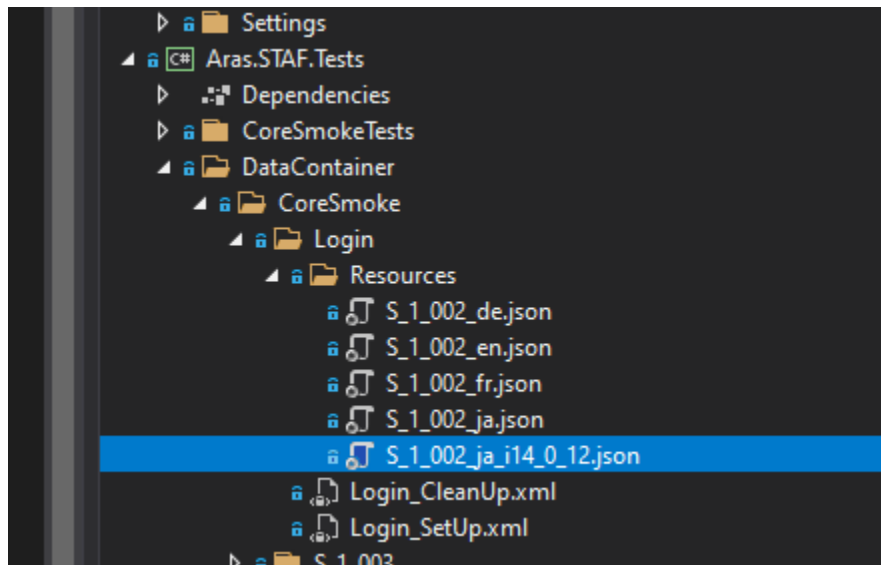


Figure 23. S_1_002_ja_i26.json

3.2.5 Updating Existing Solution created with TAF 1.29 or earlier for Aras Innovator 14 or later from External Innovator versioning approach to Internal one

Note: Without this update, all Selenium tests created with TAF 1.29 or earlier will be executed but we recommend making those changes to simplify future TAF updates.

1. Install the TAF 1.32 Template (NuGet for VS2022 or VSIX for VS2019) – see section 3.1 of Upgrade Guide.
2. Create a new solution using the **Aras Innovator 14+ Selenium Tests Template** – see section 7.1 of Installation Guide.
3. In the Visual Studio Solution Explorer, double-click the **AppConfigValidation.xsd** file to open it.
4. In the **AppConfigValidation.xsd**, update the “InnovatorVersionList” tag in your existing solution with the values from the “InnovatorVersionList” section from the newly created solution.

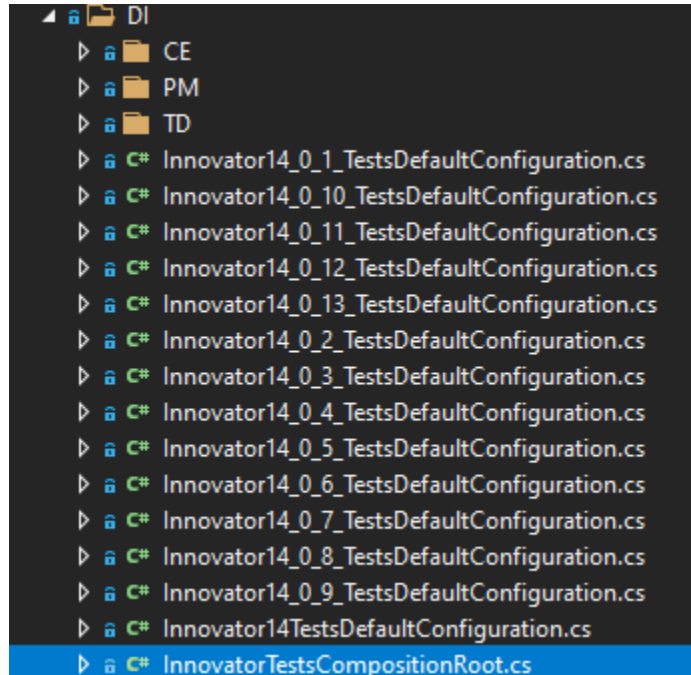
```

25 + </xsd:simpleType>
26
27 + <xsd:simpleType name="InnovatorVersionList">
28 + + <xsd:restriction base="xsd:string">
29 + + + <xsd:enumeration value="14.0"/>
30 + + + <xsd:enumeration value="14.0.1"/>
31 + + + <xsd:enumeration value="14.0.2"/>
32 + + + <xsd:enumeration value="14.0.3"/>
33 + + + <xsd:enumeration value="14.0.4"/>
34 + + + <xsd:enumeration value="14.0.5"/>
35 + + + <xsd:enumeration value="14.0.6"/>
36 + + + <xsd:enumeration value="14.0.7"/>
37 + + + <xsd:enumeration value="14.0.8"/>
38 + + + <xsd:enumeration value="14.0.9"/>
39 + + + <xsd:enumeration value="14.0.10"/>
40 + + + <xsd:enumeration value="14.0.11"/>
41 + + + <xsd:enumeration value="14.0.12"/>
42 + + </xsd:restriction>
43 + </xsd:simpleType>
44

```

Figure 24. AppConfigValidation.xsd

- In the “DI” folder of your existing solution, replace the files ending in *TestsDefaultConfiguration.cs for Aras Innovator versions 15 through 26 with the *TestsDefaultConfiguration.cs files for versions 14_0_1 through 14_0_12 from the newly created solution.



- Update the existing rebinds in the *TestsDefaultConfigurations.cs files to the corresponding new *TestsDefaultConfiguration.cs files.

Rebinds from “Innovator15TestsDefaultConfiguration” should be moved to “Innovator14_0_1_TestsDefaultConfiguration”, rebinds from “Innovator16TestsDefaultConfiguration” should be moved to “Innovator14_0_2_TestsDefaultConfiguration”, etc.

- Replace the rebind keys and values of the “InnovatorVersionForTestsModulesMap” dictionary found in the “InnovatorTestsCompositionRoot” class of your existing Solution with the rebind keys and values from the “InnovatorVersionForTestsModulesMap” dictionary of the newly created solution.

```

public InnovatorTestsCompositionRoot(IIocSettings settings) : base(settings)
{
    InnovatorVersionForTestsModulesMap =
        new Dictionary<InnovatorVersion, INinjectModule>
        {
            [InnovatorVersion.I14] = new Innovator14TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_1] = new Innovator14_0_1_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_2] = new Innovator14_0_2_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_3] = new Innovator14_0_3_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_4] = new Innovator14_0_4_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_5] = new Innovator14_0_5_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_6] = new Innovator14_0_6_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_7] = new Innovator14_0_7_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_8] = new Innovator14_0_8_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_9] = new Innovator14_0_9_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_10] = new Innovator14_0_10_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_11] = new Innovator14_0_11_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_12] = new Innovator14_0_12_TestsDefaultConfiguration(),
            [InnovatorVersion.I14_0_13] = new Innovator14_0_13_TestsDefaultConfiguration()
        };
}

```