

How to Use the Aras Migration Analyzer

Setup

Setting Up the Database

Perform the following steps for each Aras Innovator database you want to analyze.

1. Unzip the Aras Migration Analyzer package. You can do this on the server or a client machine.
2. Open the Nash utility page for your Innovator installation in a web browser:
<http://{server}/{webalias}/Client/scripts/Nash.aspx>

User: Password: Database:

Password Hash Algorithm: ☒ MD5 ☐ SHA256

TimeZone: Culture:

Action: Server:

XML:

```
<AML>
<Item type="Method" id="20A077F8E6E24809908340B944423111" action="add">
  <comments />
  <execution_allowed_to keyed_name="World"
type="Identity">A73B655731924CD0B027E4F4D5FCC0A9</execution_allowed_to>
  <method_code>
    <![CDATA[
      CCO.Response.Clear();

      bool.TryParse(this.getProperty("is_readiness_check", ""), out bool isReadinessCheck);

      string tempFolderParam = serverConnection.GetOperatingParameter("ServerMethodTempDir",
      "").ToString();
      string debugEnabledParam = serverConnection.GetOperatingParameter("DebugServerMethod",
      "").ToString();
    ]]>
  </method_code>
</Item>
```

Nash.aspx

3. Enter credentials for an administrative user like admin or root and click **Login**.
4. In the Aras Migration Analyzer package, open **InnovatorSideUtility/AMA_UTILITYMethod.xml**.
5. Copy the contents of AMA_UTILITYMethod.xml and paste into the **XML** field in the Nash form.
6. Click **Submit** to add the method to the Aras Innovator database you want to analyze.

Important



Executing this AML more than once will result in a “Method already exists” error message. If this happens, move on to the next section.

Setting Up the Code Tree

Perform the following steps for each Aras Innovator installation you want to analyze.

1. Open the **InnovatorServerConfig.xml** file in the installation directory of your Aras Innovator instance.
2. Add the following line to the file if it does not exist. If the DebugServerMethod operating parameter does exist, make sure it is set to “true”.
3. Save the InnovatorServerConfig.xml file.

Important

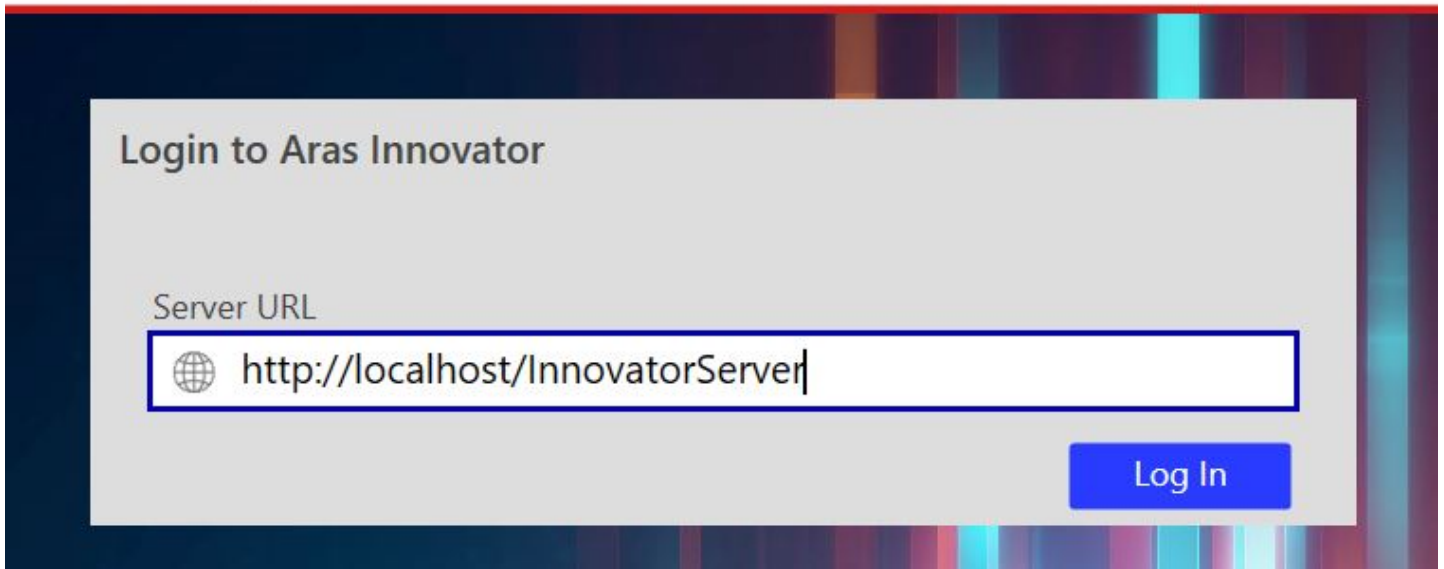
After you finish running the analysis, be sure to come back to this file and set the DebugServerMethod operating parameter to “false”. Leaving the parameter enabled for long periods of time cause many log files to accumulate on the server.

Running the Method Code Analysis

1. In the Aras Migration Analyzer package, find **MigrationAnalyzer.WpfUI.exe** and run it to start the Aras Migration Analyzer.
2. Enter the url of your Aras Innovator instance and click **Login**.



aras // MIGRATION ANALYZER



The image shows a login window titled "Login to Aras Innovator". It features a "Server URL" label above a text input field. The input field contains a globe icon followed by the text "http://localhost/InnovatorServer". A blue "Log In" button is positioned to the right of the input field. The background of the window is dark blue with vertical light blue and white stripes.

Aras Migration Analyzer login page

3. When the Aras Innovator login page appears, choose the database you want to analyze and enter credentials for an administrative user like admin or root. Click **Login**.

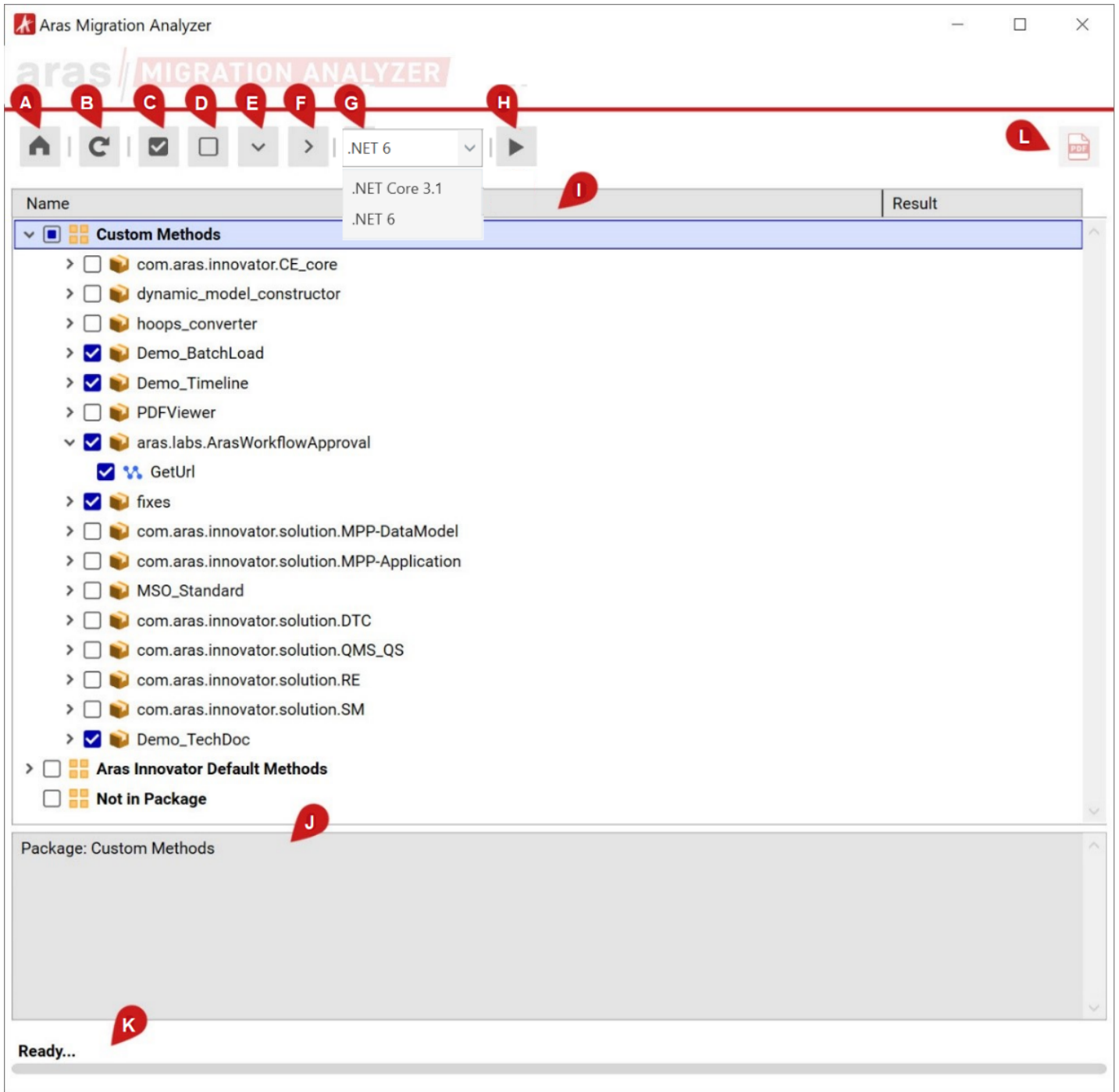
aras INNOVATOR

Release 35 Build 14.35.0.43813



Aras Innovator login page

4. Next, the tool will present the main page for running the method code analysis.



Main page for choosing methods to analyze

1. **Home:** Return to the initial login page.
2. **Refresh:** Reload the method list from the Aras Innovator database.

3. **Select All:** Select every method in the list.
4. **Deselect All:** Deselect every method in the list.
5. **Expand All:** Expand all categories and packages to see every method in the list.
6. **Collapse All:** Collapse all categories to hide all packages and methods in the list.
7. **.Net Version:** Choose the .Net version (.NET 6 or .NET Core 3.1) to check the server methods for compatibility analysis.
8. **Analyze:** Run the .NET Core migration analysis on the selected methods.
9. **Method List:** This pane contains a list of server methods in the Aras Innovator database, organized by package. Packages are grouped into three categories:
 1. **Custom Methods:** Contains all packages that are not core Aras Innovator packages. This includes Aras applications and custom packages.
 2. **Aras Innovator Default Methods:** Contains core Aras Innovator packages. This category is deselected by default because Aras migrated all core code to .NET Core when subscribers upgraded to Release 15.
 3. **Not in Package:** Contains methods that don't belong to a package definition.
10. **Detail Pane:** Presents detailed analysis information about the selected method.
11. **Status Bar:** Displays the progress of the analysis. When the analysis is complete, the status bar indicates how many methods are compatible with .NET Core, incompatible, or have compilation errors.
12. **Export to PDF:** Allows the user to save the analysis results to a PDF file for review and sharing.
13. Select the methods you want to analyze from the method list and click the **Analyze** button.
14. Wait until the status bar indicates that the analysis is complete. Now you're ready to review the results.



Status bar after analyzing 8 methods

Understanding the Analysis Results

There are two ways to review the results of the method analysis – inside the migration analyzer tool itself, or in an exported PDF.

Important



The Aras Migration Analyzer tool analyzes the compatibility of code only within the chosen server methods of an Aras Innovator database. It does not analyze code in custom DLLs that these server methods refer to. To assess the compatibility of code in a custom DLL, please use the relevant analyzer tool offered by Microsoft: a) The [API Port Utility](#) evaluates code for compatibility with .NET Core 3.1 b) The [.NET Upgrade Assistant](#) evaluates code for compatibility against .NET 6.

Reviewing Results in the Migration Analyzer



aras

MIGRATION ANALYZER

Home

Refresh

Check

Uncheck

Dropdown

Next

Target: .NET 6

Run

PDF

Name	Result
Custom Methods com.aras.innovator.CE_core dynamic_model_constructor hoops_converter Demo_BatchLoad Demo_BatchLoad Demo_FileReps Demo_WorkflowVote Demo_Timeline Demo_Timeline_onBeforeUpdate Demo_UpdateBaselineDate PDFViewer aras.labs.ArasWorkflowApproval GetUrl	Compatible Compatible Compatible Compatible Compatible Compatible Incompatible

Method: GetUrl (50F46B7F0F1A43369C071476ACB6769C)
Package: aras.labs.ArasWorkflowApproval

INCOMPATIBLE
'T:System.Web.HttpContext' type has incompatible members:
 .get_Current
 .get_Request
'T:System.Web.HttpRequest' type has incompatible members:

Ready... 7 of 8 compatible (88%)

0 errors (0%) 1 incompatible (12%)

Completed method analysis in the Migration Analyzer UI



Once the analysis is complete, the righthand column of the method list will indicate one of the following results for each of the selected methods.

- **Compatible:** The code in this method is compatible with .NET Core and doesn't require updates.
- **Incompatible:** One or more .NET Framework APIs used in this method does not port to .NET Core. This code will need to be updated before it will work in Aras Innovator 15+. Click the method in the list and review the details pane to learn which specific APIs will need to be replaced.
- **Error:** There was a compilation error during the analysis. Click the method in the list and review the details pane for more information about the error.

For more information about migrating .NET Framework code to .NET Core, visit Microsoft's migration guide: <https://docs.microsoft.com/en-us/dotnet/core/porting/>

Reviewing Results in a PDF

aras // MIGRATION ANALYZER

Server URL
http://localhost/12SP18

Database
Demo12SP18

Analysis Date
Wednesday, September 29, 2021

7 compatible (88%)

0 errors (0%) 1 incompatible (12%)

8 Analyzed

Name	Result	Details
 Demo_BatchLoad		
 Demo_BatchLoad	Compatible	
 Demo_FileReps	Compatible	
 Demo_WorkflowVote	Compatible	
 Demo_Timeline		
 Demo_Timeline_onBeforeUpdate	Compatible	
 Demo_UpdateBaselineDate	Compatible	
 aras.labs.ArasWorkflowApproval		
 GetUrl	Incompatible	'T:System.Web.HttpContext' type has incompatible members: .get_Current .get_Request 'T:System.Web.HttpRequest' type has incompatible members: .get_Url
 fixes		
 labs_changelidentity	Compatible	
 Demo_TechDoc		
 tp_BOMContentGenerator	Compatible	

Completed analysis saved to PDF



Once the analysis is complete, you have the option to save the results to a PDF file. The file contains all the same information as the Migration Analyzer UI, but in an easily shared format. The PDF shows a result and details (where necessary) for each analyzed method.

