

ILLINOIS PERMITTING USING THE RATING TOOL API



Ruben Boehler, PE, SE
Engineer of Structural Services
Bureau of Bridges & Structures



AASHTOWare BrR
2024 RADBUG

OVERVIEW

- AASHTOWare Load Rating Tool
 - BrR User Interface
 - How to set it up
 - How to use it
 - API Example
 - Out of the box
 - Customized

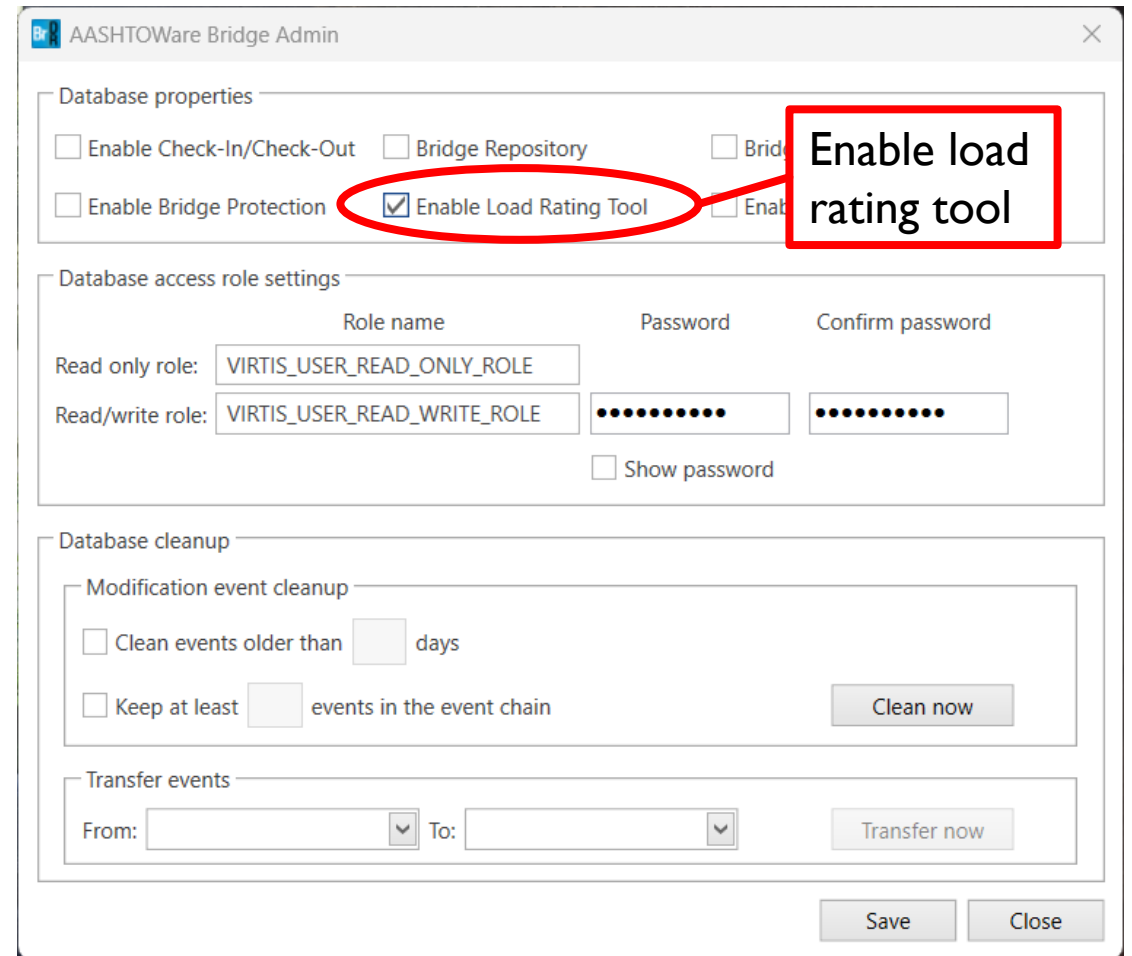
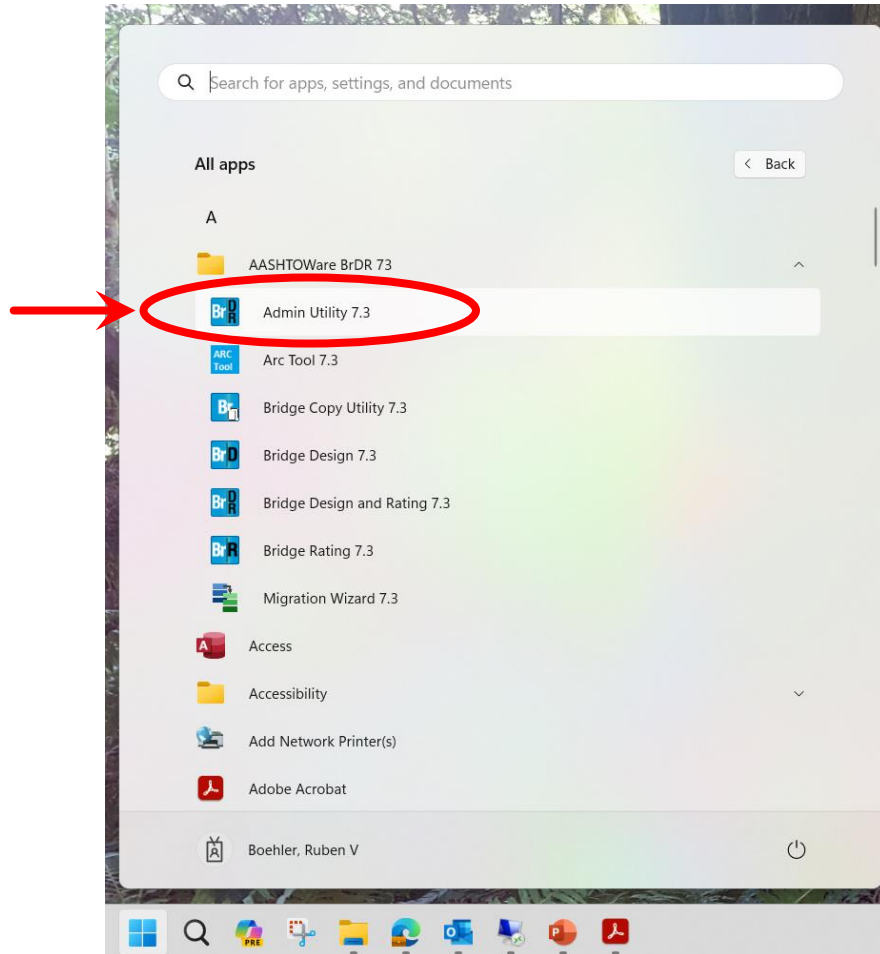
LOAD RATING TOOL

- BrR User Interface
 - manage precomputed data
 - analyze permit loads

BID	Bridge ID	Bridge Name
7367	0431113	CFS/RPW (US 20/IL 84 over Creek)(PCBC)
6467	0432001	VPT/DWT/JGT (IL 84 over Trib of Apple River)
4225	0432006	JRT/ (US 20 over Yellow Creek Trib.)(CBC)
3400	0440001	TPL/MKT (US 45 over Pond Creek)(RCS)
7076	0440004	DLG/JTB (US 45 over L. Cache Cr.)(PSD)
3539	0440005	TPL/TES (US 45 over McCorkle Cr.)(PSD)
2272	0440010	TES/MKT(US 45 over Cave Cr.)(PSD)
2451	0440011	TPL/MKT/TES (IL 37 over Cache R.)(CWF)
2452	0440012	TPL/MKT/TES (IL 37 over Aband. RR)(CWF)
2485	0440014	CWC/TES (IL 146 over Cache R.)(SWF/PSD)
698	0440022	CWC/MKT(IL 146 over Bell Pond Cr.)(PSD)
2662	0440025	TPL/MKT/JRS(IL 147 over Cedar Creek)(CWF)

STEP I – CONFIGURE

■ AASHTOWare Bridge Admin. Utility



STEP I – CONFIGURE

System Defaults x

General Bridge workspace Superstructure analysis Specifications Substructure analysis Tolerance Custom agency fields Rating tool

Load rating tool repository path:

C:\rvb\AASHTOWareBr73-LRT-Repository\3A15115C-19BE-4085-A001-B79F8FD6E102-7.3.0.3001

Processing order	Code	Description	Pass condition	% impact (%)	One lane restriction
1	2	Pass, no restrictions		100	<input type="checkbox"/>
2	4	Pass, reduced speed (45 mph or less)		33	<input type="checkbox"/>
3	3	Pass, one lane restriction - No Other Vehicle on Bridge		100	<input checked="" type="checkbox"/>
4	6	Pass, one lane and reduced speed (45 mph or less)		33	<input checked="" type="checkbox"/>
5	5	Pass, crawl speed (5 mph or less)		0	<input type="checkbox"/>
6	7	Pass, one lane and crawl speed (5 mph or less)		0	<input checked="" type="checkbox"/>

Specify where to save Precomputed Data Files

Specify:
Processing Order
Code to Return
% of Impact
Lane Restrictions

Denied code:

Not rated code:

STEP 1 – CONFIGURE

- Configuration has been Completed
- Only need to do this once
- Ready to Create Precomputed Data Files

STEP 2 – GENERATE PRECOMPUTED DATA

The screenshot shows the AASHTOWare Bridge Rating software interface. The 'RATE' ribbon is active, displaying various tools. A red box highlights the 'Precomputed Data' button, which is labeled '01 10'. A red arrow points from this button to a text box that says 'Click Precomputed Data Button'. On the left, the 'BRIDGE EXPLORER' pane shows a tree view of bridge folders. A red box highlights the 'All Bridges' folder, with a red arrow pointing to a text box that says 'Select Model(s)'. The main table displays a list of bridges with columns for BID, Bridge ID, and Bridge Name. The row with BID 2451 and Bridge ID 0440011 is highlighted in yellow.

BID	Bridge ID	Bridge Name
7367	0431113	CFS/RPW (US 20/IL 84 over Creek)(PCBC)
6467	0432001	VPT/DWT/JGT (IL 84 over Trib of Apple River)
4225	0432006	JRT/ (US 20 over Yellow Creek Trib.) (CBC)
3400	0440001	TPL/MKT (US 45 over Pond Creek)(RCS)
7076	0440004	DLG/JTB (US 45 over L. Cache Cr.)(PSD)
3539	0440005	TPL/TES (US 45 over McCorkle Cr.)(PSD)
2272	0440010	TES/MKT(US 45 over Cave Cr.)(PSD)
2451	0440011	TPL/MKT/TES (IL 37 over Cache R.)(CWF)
2452	0440012	TPL/MKT/TES (IL 37 over Aband. RR)(CWF)
2485	0440014	CWC/TES (IL 146 over Cache R.)(SWF/PSD)
698	0440022	CWC/MKT(IL 146 over Bell Pond Cr.)(PSD)
2662	0440025	TPL/MKT/JRS(IL 147 over Cedar Creek)(CWF)

Click Precomputed Data Button

Select Model(s)

STEP 2 – GENERATE PRECOMPUTED DATA

Precomputed Data

Generate Maintain

Analysis type: Line Girder
Rating method: LFD

Points of interest

Override bridge points of interest

Steel member

- Generate at tenth points
- Generate at section change points
- Generate at user defined points

Concrete member

- Generate at tenth points except supports
- Generate at support points
- Generate at support face & critical shear points
- Generate at section change points
- Generate at user-defined points

Overwrite existing precomputed data
 Stop on first error

Save as system defaults Generate

Generate Precomputed Data

Close

Analysis Progress

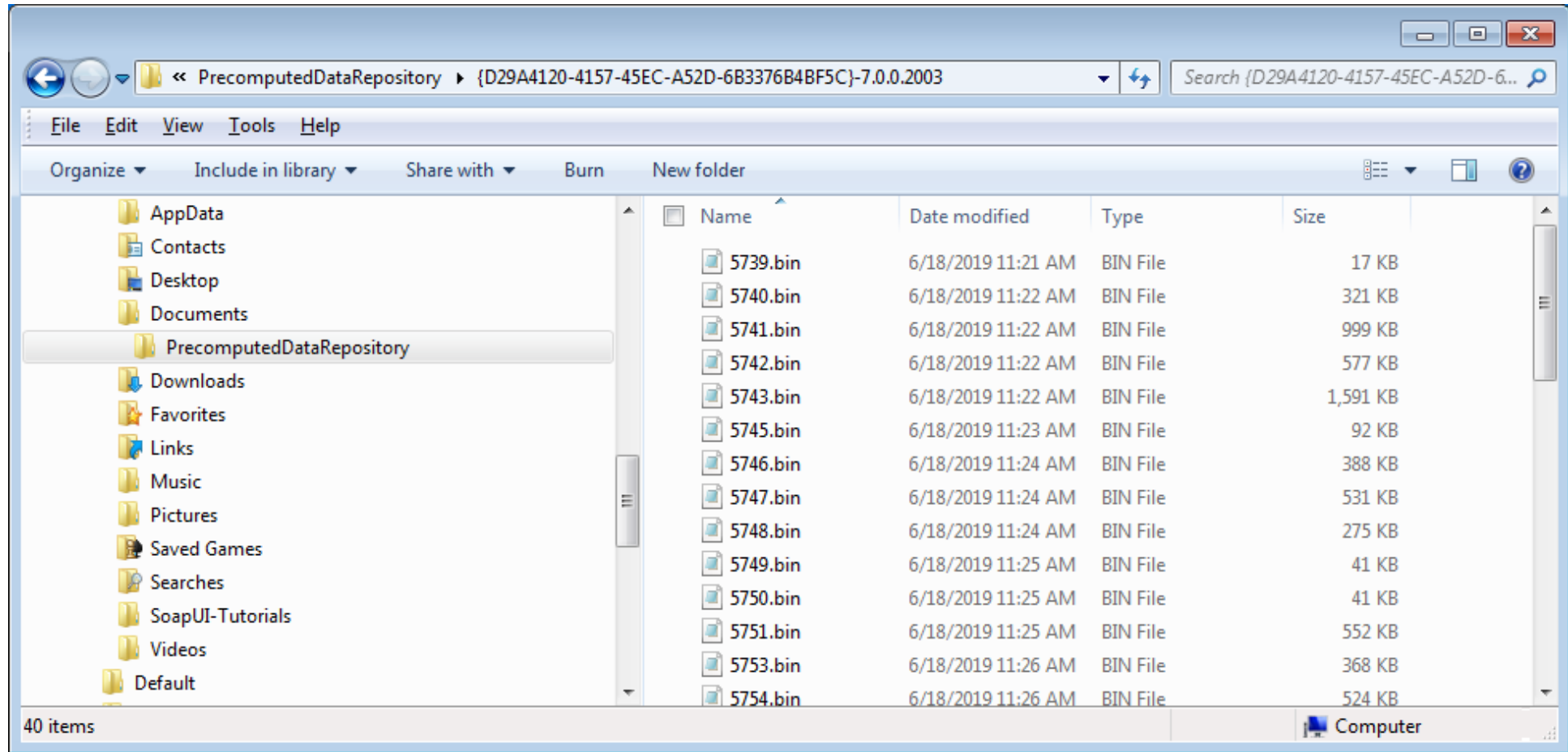
- Analysis Event
 - 0440011
 - STRUCTURES
 - Sp. 1-5 [Sp 1-5 (5-Sp. Cont.) WF-x]
 - GIRDER-SYSTEM MEMBERS
 - 1 - E Fascia {Existing Membe
 - 2 - 1st E Int {Existing Membe
 - 3 - 2nd E Int {Existing Memb
 - 4 - 3rd E Int-x [W24x62-Cor
 - 5 - 3rd W Int [W24x62-Com
 - 6 - 2nd W Int {Existing Mem
 - 7 - 1st W Int {Existing Memb
 - 8 - W Fascia {Existing Memb

- Location - 154.1667 (ft)
- Location - 159.1667 (ft)
- Location - 159.5000 (ft)
- Location - 164.1667 (ft)
- Location - 169.1667 (ft)
- Location - 174.1667 (ft)
- Location - 179.1667 (ft)
- Location - 184.1667 (ft)
- Location - 186.0000 (ft)
- Location - 189.1667 (ft)
- Location - 194.1667 (ft)
- Location - 199.1667 (ft)
- Location - 204.2417 (ft)
- Location - 209.2709 (ft)
- Location - 209.3167 (ft)
- Location - 214.3917 (ft)
- Location - 219.4667 (ft)
- Location - 224.5417 (ft)
- Location - 229.6167 (ft)
- Location - 234.6917 (ft)
- Location - 239.7667 (ft)
- Location - 244.8417 (ft)
- Location - 249.9167 (ft)

Completed Specification Check.
Info - LFR analysis successfully completed!
Info - Analysis completed!

Finished processing 1 of 1 bridges: OK Cancel

STEP 2 – GENERATE PRECOMPUTED DATA



STEP 3 – ANALYZE PERMIT VEHICLE

Minimum Allowable Rating Factor

Select Permit Vehicle

List of Bridges to be Analyzed

Load Rating Tool

Permit application number:

Application date: 7/20/2024

Requested by:

Minimum allowable rating factor: 1.00

Comment:

Bridges Vehicles Rating results

Configure analysis settings...

BID	Bridge database					Has precomputed data	Travel direction
	Bridge ID	NBI structure ID	Route number	Number of structures	Completely defined		
49	0840017	084-0017	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
59	0540042	054-0042	10055	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
60	0540043	054-0043	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
64	0540047	054-0047	10055	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
66	0540053	054-0053	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
68	0540055	054-0055	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
74	0540057	054-0057	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
77	0540061	054-0061	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
78	0540063	054-0063	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
80	0540065	054-0065	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
90	0570155	057-0155	10055	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
93	0570007	057-0007	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
102	0570016	057-0016	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions
104	0570022	057-0022	10055	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Both directions

Process permit...

STEP 3 – ANALYZE PERMIT VEHICLE

Load Rating Tool

Permit application number: [input field]

Application date: 7/22/2024

Comment: [input field]

Bridges Vehicles Rating results

Filter results: Pass Fail Exceptions

Vehicle	BID	Bridge ID	Route number	Code	Description	LFD		LRFR Permit rating factor	Controlling impact	Pass conditions	Analysis warnings
						Inventory rating factor	Operating rating factor				
permit...	49	0840017	10055	1	Denied		0.924		1.000		
permit...	49	0840017	10055	4	Pass, with reduced spe...		1.062		0.330		
permit...	59	0540042	10055	2	Pass, with no restrictions		1.104		1.000		
permit...	60	0540043	10055	1	Denied		0.704		1.000		
permit...	60	0540043	10055	1	Denied		0.817		0.330		
permit...	60	0540043	10055	1	Denied		0.896		1.000		
permit...	60	0540043	10055	6	Pass, with one lane an...		1.040		0.330		
permit...	64	0540047	10055	2	Pass, with no restrictions		1.371		1.000		
permit...	66	0540053	10055	2	Pass, with no restrictions		1.131		1.000		
permit...	68	0540055	10055	2	Pass, with no restrictions		1.025		1.000		
permit...	74	0540057	10055	2	Pass, with no restrictions		1.853		1.000		
permit...	77	0540061	10055	2	Pass, with no restrictions		1.556		1.000		
permit...	78	0540063	10055	2	Pass, with no restrictions		1.188		1.000		
permit...	80	0540065	10055	2	Pass, with no restrictions		1.586		1.000		

Create rating results file... View rating results file...

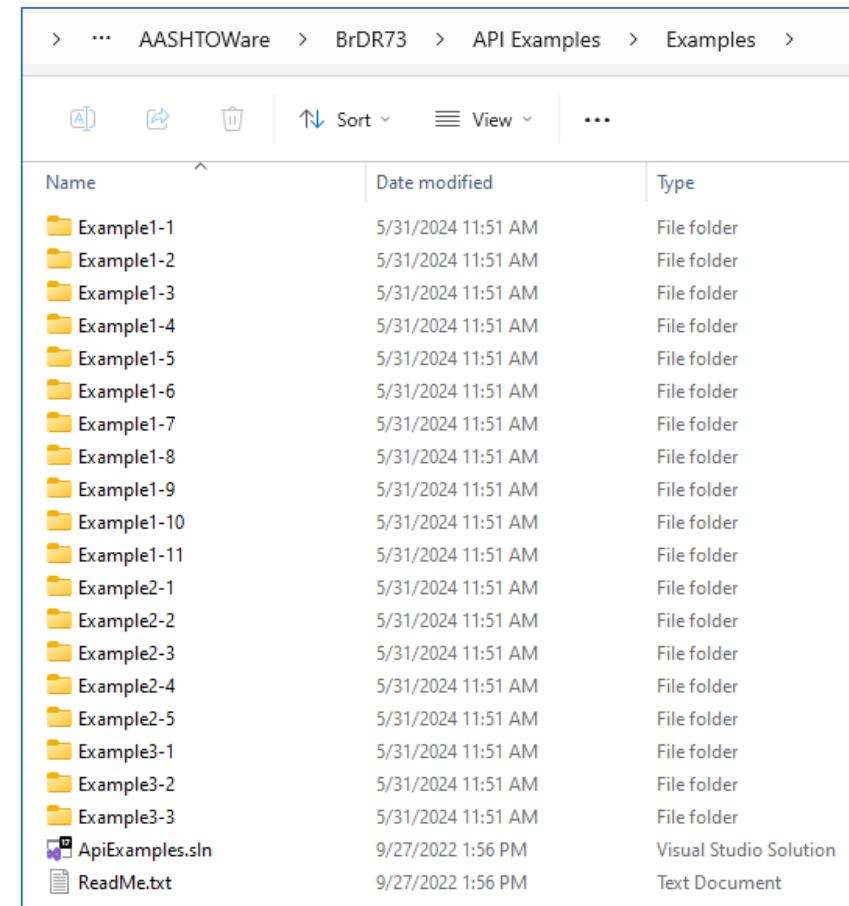
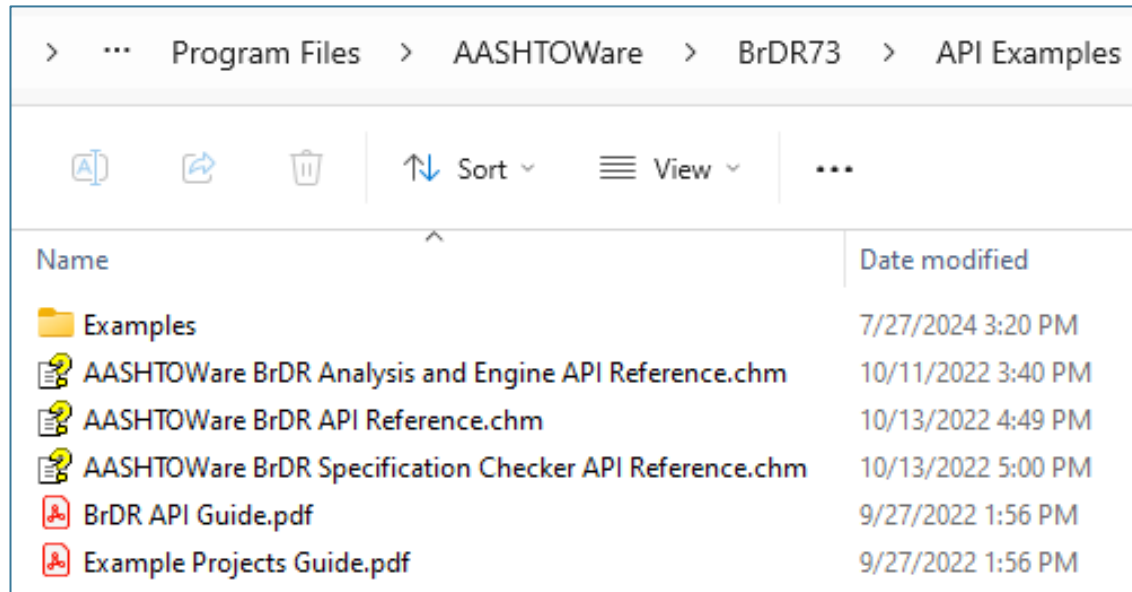
Code set during Configuration

≈ 11 seconds

Operating Rating Factor

BrDR – API EXAMPLES

- C:\Program Files\AASHTOWare\BrDR73\API Examples
- 3 References and 2 Guides
- 19 Code Examples



BrDR – API EXAMPLES

■ 19 Examples

1 Basic API Operations

- 1.1 Object retrieve examples
- 1.2 Object CRUD examples
- 1.3 Summary object examples
- 1.4 Retrieving reinforced concrete cross sections
- 1.5 Retrieving prestressed beam spans
- 1.6 C++ Object retrieve example (a subset of Example 1.1) [Not in solution]
- 1.7 Calculating Steel Cross-Sectional Area
- 1.8 Create and save bridge example
- 1.9 Import Utility example
- 1.10 Licensing example
- 1.11 IFC bridge import/export example

2 Analysis Examples

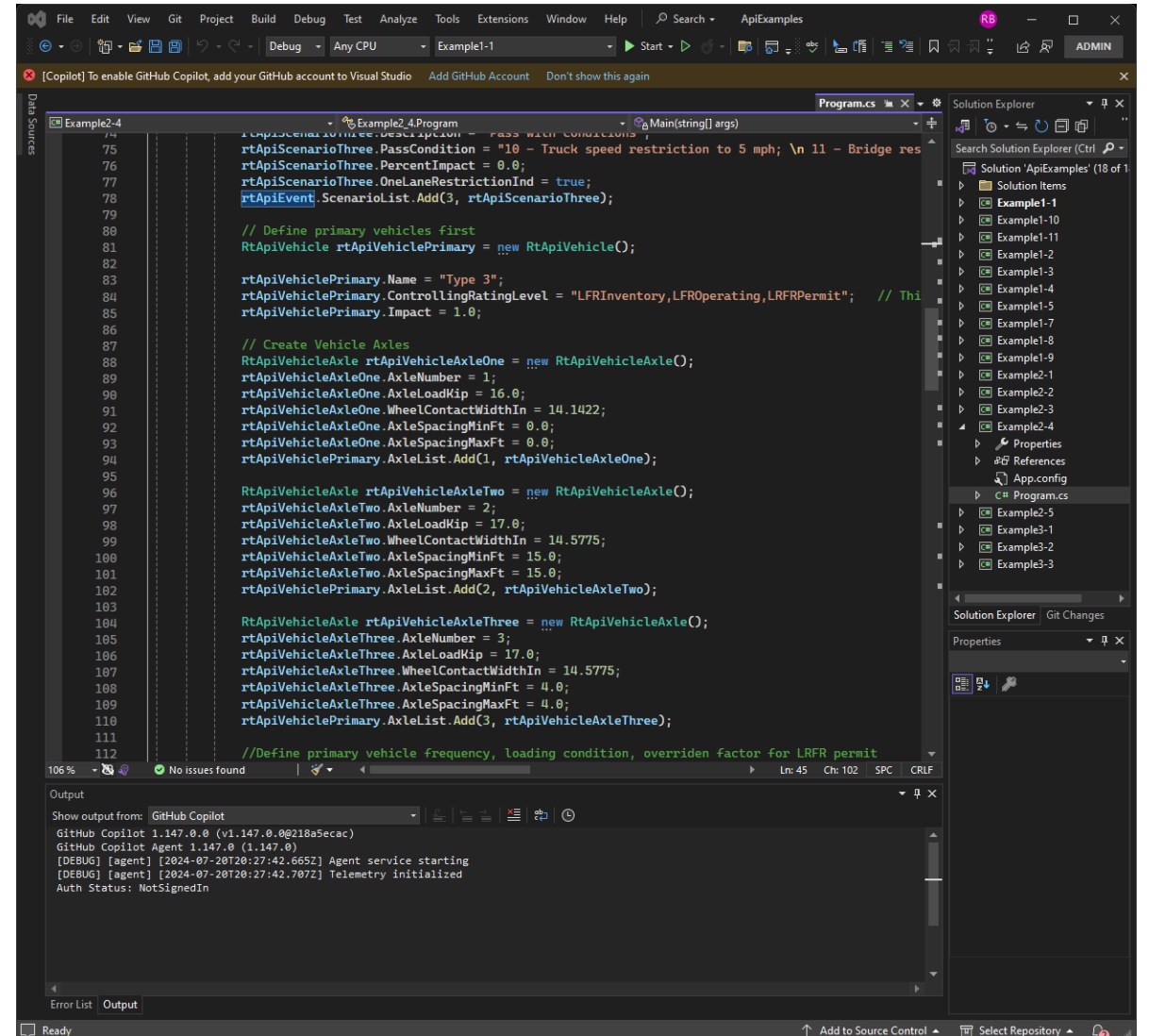
- 2.1 Analysis API
- 2.2 Simplified Analysis API (LFD)
- 2.3 XML-based Analysis API
- 2.4 Rating Tool API ←
- 2.5 Simplified Analysis API (LRFR)

3 Engine

- 3.1 Creating a 3rd-party engine
- 3.2 Creating a 3rd-party engine properties UI window
- 3.3 Retrieve Engine Properties from Bridge Object

RATING TOOL - API EXAMPLE

- Example 2-4
- .NET Framework 4.8
- C#
- Default Scenarios
- Type 3 and EV2 Vehicles
- Set-up to work with Default Bridge Models



```
74 rtApiScenarioThree.Description = "Pass with conditions";
75 rtApiScenarioThree.PassCondition = "10 - Truck speed restriction to 5 mph; \n 11 - Bridge res
76 rtApiScenarioThree.PercentImpact = 0.0;
77 rtApiScenarioThree.OneLaneRestrictionInd = true;
78 rtApiEvent.ScenarioList.Add(3, rtApiScenarioThree);
79
80 // Define primary vehicles first
81 RtApiVehicle rtApiVehiclePrimary = new RtApiVehicle();
82
83 rtApiVehiclePrimary.Name = "Type 3";
84 rtApiVehiclePrimary.ControllingRatingLevel = "LFRInventory, LFR0perating, LFRFRpermit"; // Thi
85 rtApiVehiclePrimary.Impact = 1.0;
86
87 // Create Vehicle Axles
88 RtApiVehicleAxle rtApiVehicleAxleOne = new RtApiVehicleAxle();
89 rtApiVehicleAxleOne.AxleNumber = 1;
90 rtApiVehicleAxleOne.AxleLoadKip = 16.0;
91 rtApiVehicleAxleOne.WheelContactWidthIn = 14.1422;
92 rtApiVehicleAxleOne.AxleSpacingMinFt = 0.0;
93 rtApiVehicleAxleOne.AxleSpacingMaxFt = 0.0;
94 rtApiVehiclePrimary.AxleList.Add(1, rtApiVehicleAxleOne);
95
96 RtApiVehicleAxle rtApiVehicleAxleTwo = new RtApiVehicleAxle();
97 rtApiVehicleAxleTwo.AxleNumber = 2;
98 rtApiVehicleAxleTwo.AxleLoadKip = 17.0;
99 rtApiVehicleAxleTwo.WheelContactWidthIn = 14.5775;
100 rtApiVehicleAxleTwo.AxleSpacingMinFt = 15.0;
101 rtApiVehicleAxleTwo.AxleSpacingMaxFt = 15.0;
102 rtApiVehiclePrimary.AxleList.Add(2, rtApiVehicleAxleTwo);
103
104 RtApiVehicleAxle rtApiVehicleAxleThree = new RtApiVehicleAxle();
105 rtApiVehicleAxleThree.AxleNumber = 3;
106 rtApiVehicleAxleThree.AxleLoadKip = 17.0;
107 rtApiVehicleAxleThree.WheelContactWidthIn = 14.5775;
108 rtApiVehicleAxleThree.AxleSpacingMinFt = 4.0;
109 rtApiVehicleAxleThree.AxleSpacingMaxFt = 4.0;
110 rtApiVehiclePrimary.AxleList.Add(3, rtApiVehicleAxleThree);
111
112 //Define primary vehicle frequency, loading condition, overridden factor for LRRF permit
```

Output

```
Show output from: GitHub Copilot
GitHub Copilot 1.147.0.0 (v1.147.0.0@218a5ecac)
GitHub Copilot Agent 1.147.0 (1.147.0)
[DEBUG] [agent] [2024-07-20T20:27:42.665Z] Agent service starting
[DEBUG] [agent] [2024-07-20T20:27:42.707Z] Telemetry initialized
Auth Status: NotSignedIn
```

RATING TOOL - API EXAMPLE

■ Before Running

- Create Precomputed Data files for
 - Must uncheck “Template” toggle in model
 - TrainingBridge1
 - Example7
 - RCTrainingBridge1
- Change Repository Path in Code

TrainingBridge1

Template

Bridge completely defined

Superstructures

Culverts

Substructures

al reference point

Traffic

Custom agency fields

Year built: 1999

Length: 161.000656 ft

Route number: 0051

Mi. post: 17

```
Program.cs* [X]
C# Example2-4
41
42 // Create event
43 RtApiEvent rtApiEvent = new RtApiEvent();
44
45 rtApiEvent.RepositoryPath = "D:\\TempRepo\\BFC6084A-2F0E-403C-8DE9-8BA9A5CB263C-7.2.0.1";
46
47 rtApiEvent.DeniedCode = "X";
48 rtApiEvent.NotRatedCode = "NA";
49
50 rtApiEvent.MinimumAllowableRatingFactor = 1.0;
```


RATING TOOL - API EXAMPLE

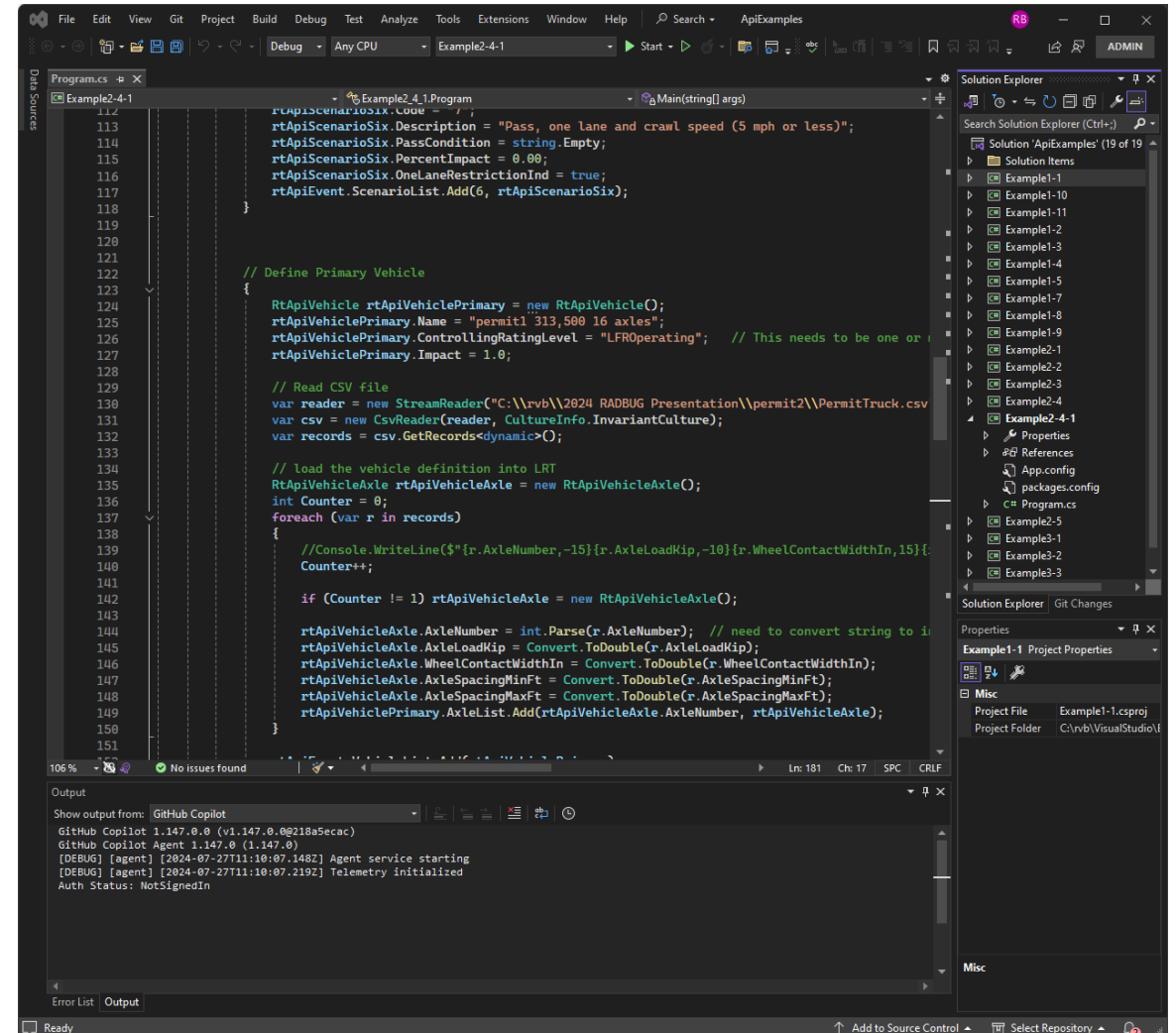
Run API

The screenshot shows the Visual Studio IDE with the 'Start' button in the Debug toolbar highlighted in red. Below the IDE, a console window displays the output of the application, which is a table of API data. The table has the following structure:

Bridge ID	Code	Description	Inventory rating factor	Operating rating factor	Permit rating factor
TrainingBridge1	1	Pass, no restrictions	1.32	2.204	∞
Example7	X	Denied	0.881	1.471	∞
Example7	2	Pass with conditions	1.063	1.775	∞
RCTrainingBridge1	1	Pass, no restrictions	1.286	2.148	∞

RATING TOOL - API EXAMPLE - CUSTOMIZED

- Modify Example
- .NET Framework 4.8
- C#
- Custom Scenarios
- Custom Vehicle
- Custom List of Bridges



The screenshot displays the Visual Studio IDE with a C# program named 'Example2-4-1'. The code is written in a dark theme and includes the following key sections:

```
rtApiScenarioSix.Code = "...";
rtApiScenarioSix.Description = "Pass, one lane and crawl speed (5 mph or less)";
rtApiScenarioSix.PassCondition = string.Empty;
rtApiScenarioSix.PercentImpact = 0.00;
rtApiScenarioSix.OneLaneRestrictionInd = true;
rtApiEvent.ScenarioList.Add(6, rtApiScenarioSix);

// Define Primary Vehicle
RtApiVehicle rtApiVehiclePrimary = new RtApiVehicle();
rtApiVehiclePrimary.Name = "permit1 313,500 16 axles";
rtApiVehiclePrimary.ControllingRatingLevel = "LFRoperating"; // This needs to be one or
rtApiVehiclePrimary.Impact = 1.0;

// Read CSV file
var reader = new StreamReader("C:\\rvb\\2024 RADBUG Presentation\\permit2\\PermitTruck.csv");
var csv = new CsvReader(reader, CultureInfo.InvariantCulture);
var records = csv.GetRecords<dynamic>();

// load the vehicle definition into LRT
RtApiVehicleAxle rtApiVehicleAxle = new RtApiVehicleAxle();
int Counter = 0;
foreach (var r in records)
{
    //Console.WriteLine($"{r.AxleNumber,-15}{r.AxleLoadKip,-10}{r.WheelContactWidthIn,15}");
    Counter++;

    if (Counter != 1) rtApiVehicleAxle = new RtApiVehicleAxle();

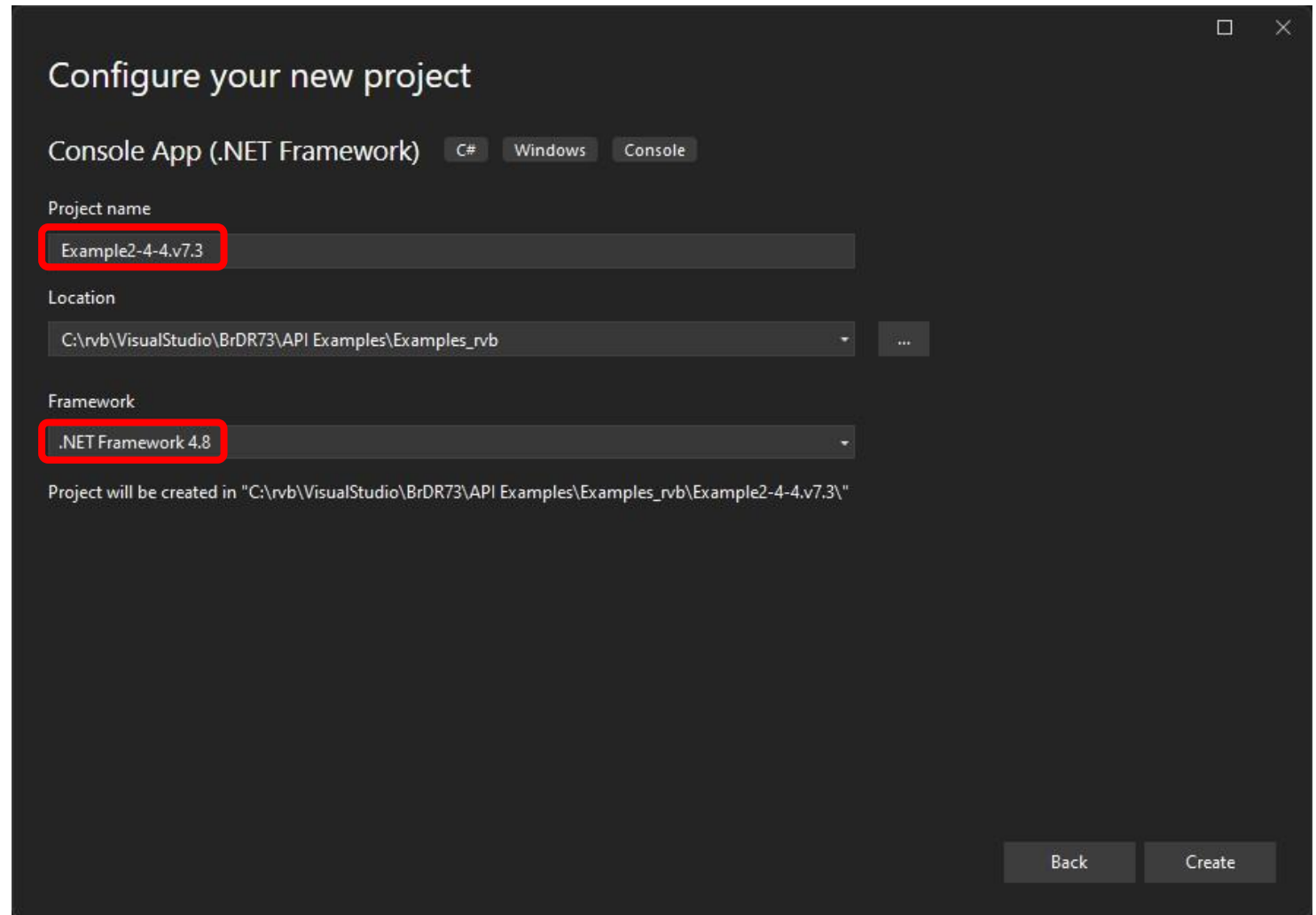
    rtApiVehicleAxle.AxleNumber = int.Parse(r.AxleNumber); // need to convert string to int
    rtApiVehicleAxle.AxleLoadKip = Convert.ToDouble(r.AxleLoadKip);
    rtApiVehicleAxle.WheelContactWidthIn = Convert.ToDouble(r.WheelContactWidthIn);
    rtApiVehicleAxle.AxleSpacingMinFt = Convert.ToDouble(r.AxleSpacingMinFt);
    rtApiVehicleAxle.AxleSpacingMaxFt = Convert.ToDouble(r.AxleSpacingMaxFt);
    rtApiVehiclePrimary.AxleList.Add(rtApiVehicleAxle.AxleNumber, rtApiVehicleAxle);
}
```

The interface also shows the Solution Explorer on the right with a tree view of project files, including 'Example2-4-1', 'Properties', 'References', 'App.config', 'packages.config', and 'C# Program.cs'. The Output window at the bottom shows logs from GitHub Copilot and the application agent.

RATING TOOL - API EXAMPLE - CUSTOMIZED

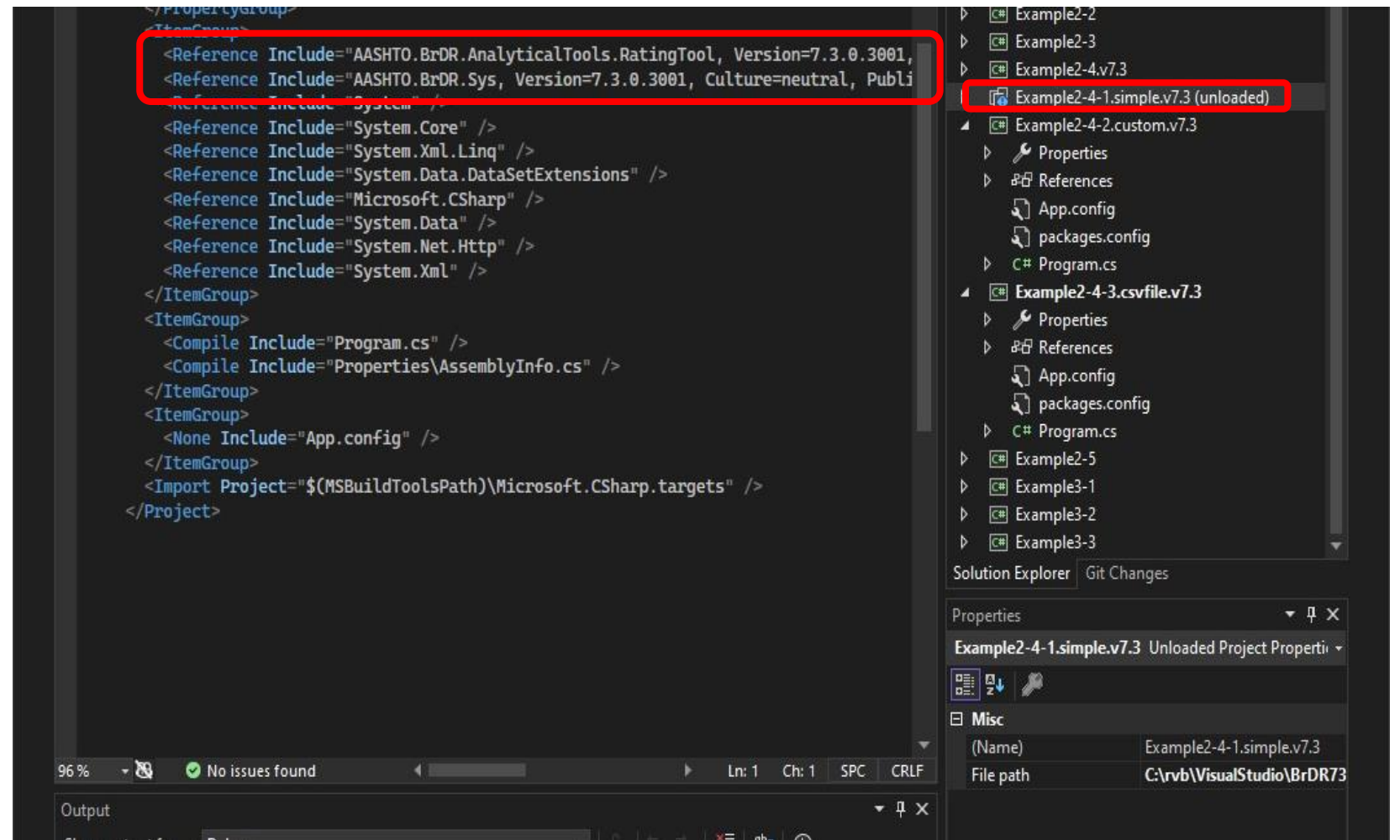
■ Create a New Project

1. Right click on Solution
>> Add
>> New Project...
2. Console App
(.NET Framework 4.8)
3. Use Unique Project Name



RATING TOOL - API EXAMPLE - CUSTOMIZED

4. Right click on the new Project and >>Unload Project
5. Copy AASHTO.BrDR References
6. Right click on the new Project and >>Reload Project
7. Copy code from example



RATING TOOL - API EXAMPLE - CUSTOMIZED

- Customize Scenarios to match IDOT's

```
51  
52 // Define scenarios  
53  
54  
55 RtApiScenario rtApiScenarioOne = new RtApiScenario();  
56 rtApiScenarioOne.ProcessOrder = 1;  
57 rtApiScenarioOne.Code = "2";  
58 rtApiScenarioOne.Description = "Pass, no restrictions";  
59 rtApiScenarioOne.PassCondition = string.Empty;  
60 rtApiScenarioOne.PercentImpact = 1.00;  
61 rtApiScenarioOne.OneLaneRestrictionInd = false;  
62 rtApiEvent.ScenarioList.Add(1, rtApiScenarioOne);  
63  
64 RtApiScenario rtApiScenarioTwo = new RtApiScenario();  
65 rtApiScenarioTwo.ProcessOrder = 2;  
66 rtApiScenarioTwo.Code = "4";  
67 rtApiScenarioTwo.Description = "Pass, reduced speed (45 mph or less)";  
68 rtApiScenarioTwo.PassCondition = string.Empty;  
69 rtApiScenarioTwo.PercentImpact = 0.33;  
70 rtApiScenarioTwo.OneLaneRestrictionInd = false;  
71 rtApiEvent.ScenarioList.Add(2, rtApiScenarioTwo);  
72  
73 RtApiScenario rtApiScenarioThree = new RtApiScenario();  
74 rtApiScenarioThree.ProcessOrder = 3;  
75 rtApiScenarioThree.Code = "3";  
76 rtApiScenarioThree.Description = "Pass, one lane restriction - No Other Vehicle on Bridge";  
77 rtApiScenarioThree.PassCondition = string.Empty;  
78 rtApiScenarioThree.PercentImpact = 1.00;  
79 rtApiScenarioThree.OneLaneRestrictionInd = true;  
80 rtApiEvent.ScenarioList.Add(3, rtApiScenarioThree);  
81  
82 RtApiScenario rtApiScenarioFour = new RtApiScenario();
```

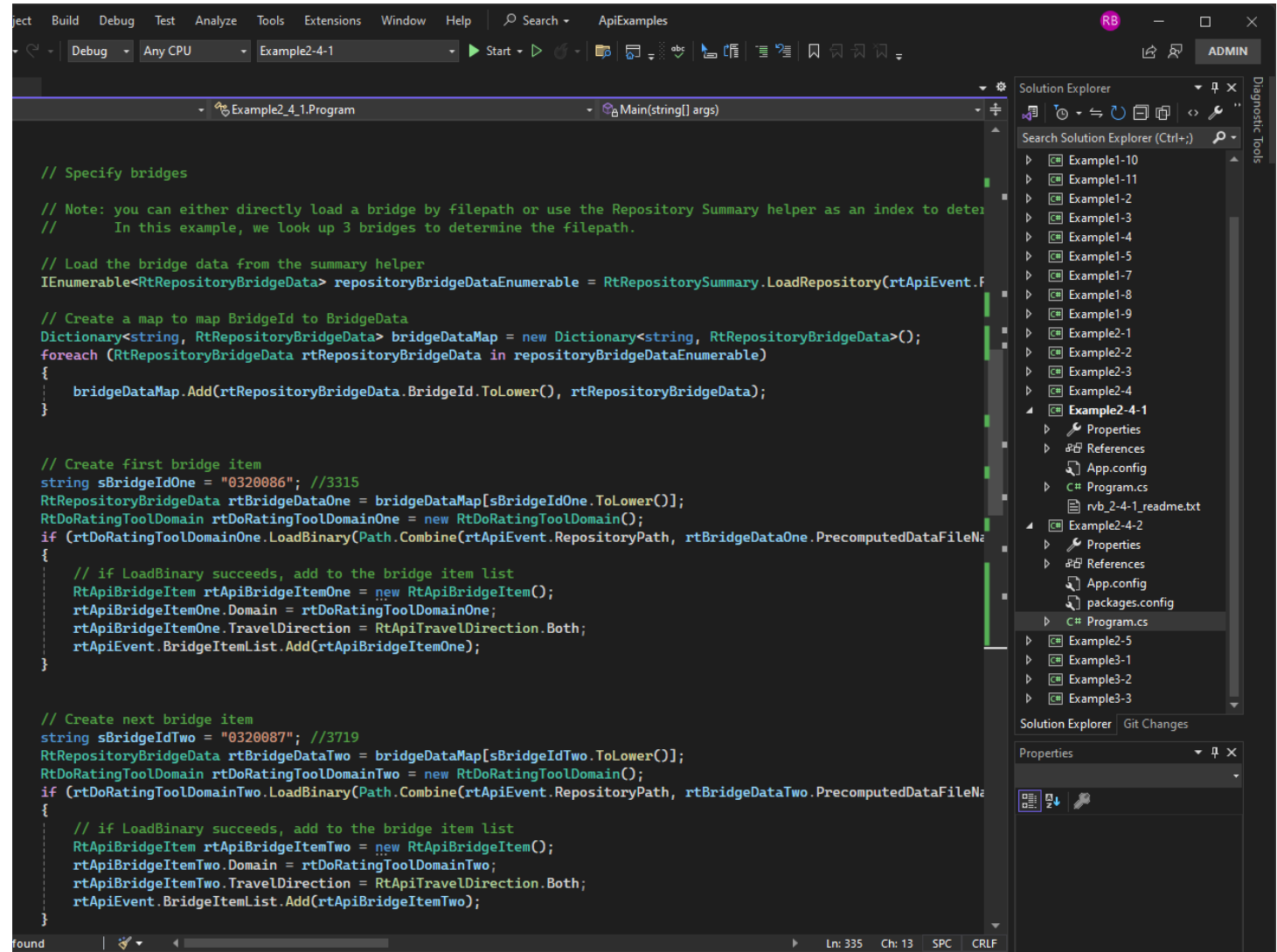
RATING TOOL - API EXAMPLE - CUSTOMIZED

- Custom Permit Vehicle
- 16 Axles
- 313,500 lbs
- 128'

```
207 RtApiVehicleAxle rtApiVehicleAxleTwelve = new RtApiVehicleAxle();
208 rtApiVehicleAxleTwelve.AxleNumber = 12;
209 rtApiVehicleAxleTwelve.AxleLoadKip = 19.7;
210 rtApiVehicleAxleTwelve.WheelContactWidthIn = 20;
211 rtApiVehicleAxleTwelve.AxleSpacingMinFt = 5;
212 rtApiVehicleAxleTwelve.AxleSpacingMaxFt = rtApiVehicleAxleTwelve.AxleSpacingMinFt;
213 rtApiVehiclePrimary.AxleList.Add(12, rtApiVehicleAxleTwelve);
214
215 RtApiVehicleAxle rtApiVehicleAxleThirteen = new RtApiVehicleAxle();
216 rtApiVehicleAxleThirteen.AxleNumber = 13;
217 rtApiVehicleAxleThirteen.AxleLoadKip = 19.7;
218 rtApiVehicleAxleThirteen.WheelContactWidthIn = 20;
219 rtApiVehicleAxleThirteen.AxleSpacingMinFt = 5;
220 rtApiVehicleAxleThirteen.AxleSpacingMaxFt = rtApiVehicleAxleThirteen.AxleSpacingMinFt;
221 rtApiVehiclePrimary.AxleList.Add(13, rtApiVehicleAxleThirteen);
222
223 RtApiVehicleAxle rtApiVehicleAxleFourteen = new RtApiVehicleAxle();
224 rtApiVehicleAxleFourteen.AxleNumber = 14;
225 rtApiVehicleAxleFourteen.AxleLoadKip = 19.7;
226 rtApiVehicleAxleFourteen.WheelContactWidthIn = 20;
227 rtApiVehicleAxleFourteen.AxleSpacingMinFt = 14.0833;
228 rtApiVehicleAxleFourteen.AxleSpacingMaxFt = rtApiVehicleAxleFourteen.AxleSpacingMinFt;
229 rtApiVehiclePrimary.AxleList.Add(14, rtApiVehicleAxleFourteen);
230
231 RtApiVehicleAxle rtApiVehicleAxleFifteen = new RtApiVehicleAxle();
232 rtApiVehicleAxleFifteen.AxleNumber = 15;
233 rtApiVehicleAxleFifteen.AxleLoadKip = 19.7;
234 rtApiVehicleAxleFifteen.WheelContactWidthIn = 20;
235 rtApiVehicleAxleFifteen.AxleSpacingMinFt = 5;
236 rtApiVehicleAxleFifteen.AxleSpacingMaxFt = rtApiVehicleAxleFifteen.AxleSpacingMinFt;
237 rtApiVehiclePrimary.AxleList.Add(15, rtApiVehicleAxleFifteen);
238
239 RtApiVehicleAxle rtApiVehicleAxleSixteen = new RtApiVehicleAxle();
240 rtApiVehicleAxleSixteen.AxleNumber = 16;
241 rtApiVehicleAxleSixteen.AxleLoadKip = 19.7;
242 rtApiVehicleAxleSixteen.WheelContactWidthIn = 20;
243 rtApiVehicleAxleSixteen.AxleSpacingMinFt = 5;
244 rtApiVehicleAxleSixteen.AxleSpacingMaxFt = rtApiVehicleAxleSixteen.AxleSpacingMinFt;
245 rtApiVehiclePrimary.AxleList.Add(16, rtApiVehicleAxleSixteen);
246
```

RATING TOOL - API EXAMPLE - CUSTOMIZED

- Custom List of Bridges
- 84 Bridges
- 8,916 PCDF's
 - 6.3 GB
- 42 seconds if Mapping 'Bridge ID' to 'BID'
- 2 seconds if Directly Loaded by File Name



```
// Specify bridges

// Note: you can either directly load a bridge by filepath or use the Repository Summary helper as an index to determine the filepath.
// In this example, we look up 3 bridges to determine the filepath.

// Load the bridge data from the summary helper
IEnumerable<RtRepositoryBridgeData> repositoryBridgeDataEnumerable = RtRepositorySummary.LoadRepository(rtApiEvent.RepositoryPath);

// Create a map to map BridgeId to BridgeData
Dictionary<string, RtRepositoryBridgeData> bridgeDataMap = new Dictionary<string, RtRepositoryBridgeData>();
foreach (RtRepositoryBridgeData rtRepositoryBridgeData in repositoryBridgeDataEnumerable)
{
    bridgeDataMap.Add(rtRepositoryBridgeData.BridgeId.ToLower(), rtRepositoryBridgeData);
}

// Create first bridge item
string sBridgeIdOne = "0320086"; //3315
RtRepositoryBridgeData rtBridgeDataOne = bridgeDataMap[sBridgeIdOne.ToLower()];
RtDoRatingToolDomain rtDoRatingToolDomainOne = new RtDoRatingToolDomain();
if (rtDoRatingToolDomainOne.LoadBinary(Path.Combine(rtApiEvent.RepositoryPath, rtBridgeDataOne.PrecomputedDataFileName)))
{
    // if LoadBinary succeeds, add to the bridge item list
    RtApiBridgeItem rtApiBridgeItemOne = new RtApiBridgeItem();
    rtApiBridgeItemOne.Domain = rtDoRatingToolDomainOne;
    rtApiBridgeItemOne.TravelDirection = RtApiTravelDirection.Both;
    rtApiEvent.BridgeItemList.Add(rtApiBridgeItemOne);
}

// Create next bridge item
string sBridgeIdTwo = "0320087"; //3719
RtRepositoryBridgeData rtBridgeDataTwo = bridgeDataMap[sBridgeIdTwo.ToLower()];
RtDoRatingToolDomain rtDoRatingToolDomainTwo = new RtDoRatingToolDomain();
if (rtDoRatingToolDomainTwo.LoadBinary(Path.Combine(rtApiEvent.RepositoryPath, rtBridgeDataTwo.PrecomputedDataFileName)))
{
    // if LoadBinary succeeds, add to the bridge item list
    RtApiBridgeItem rtApiBridgeItemTwo = new RtApiBridgeItem();
    rtApiBridgeItemTwo.Domain = rtDoRatingToolDomainTwo;
    rtApiBridgeItemTwo.TravelDirection = RtApiTravelDirection.Both;
    rtApiEvent.BridgeItemList.Add(rtApiBridgeItemTwo);
}
```

The screenshot shows a Visual Studio IDE with a C# code file open. The code is a program that loads bridge data from a repository and creates bridge items. It includes comments explaining the process of loading data and creating items. The Solution Explorer on the right shows a project structure with multiple examples and a specific example selected.

THE END

