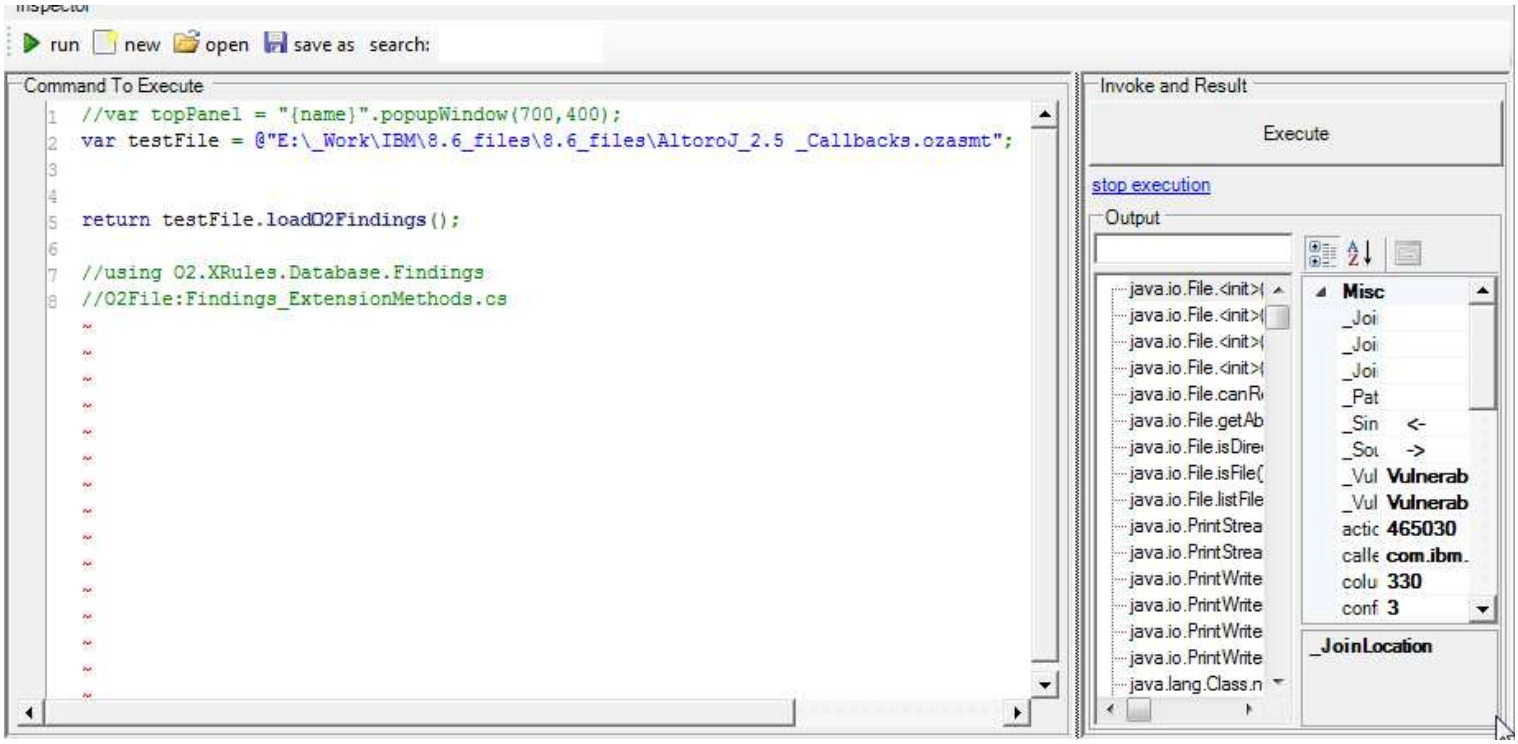
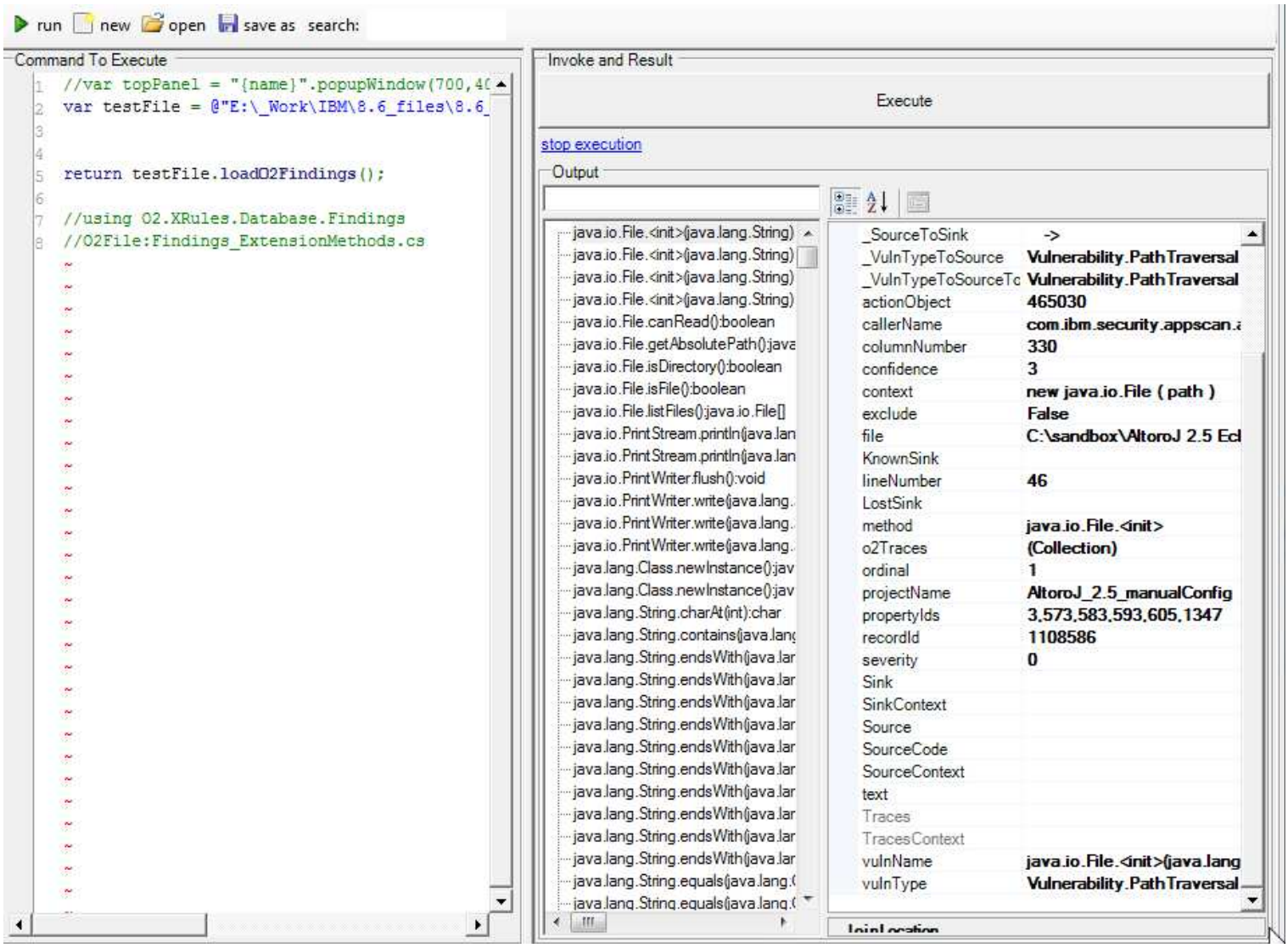
Ozasmt file format - Saving into AppScan 7.x format (Part 1)

Loading an assemment file:

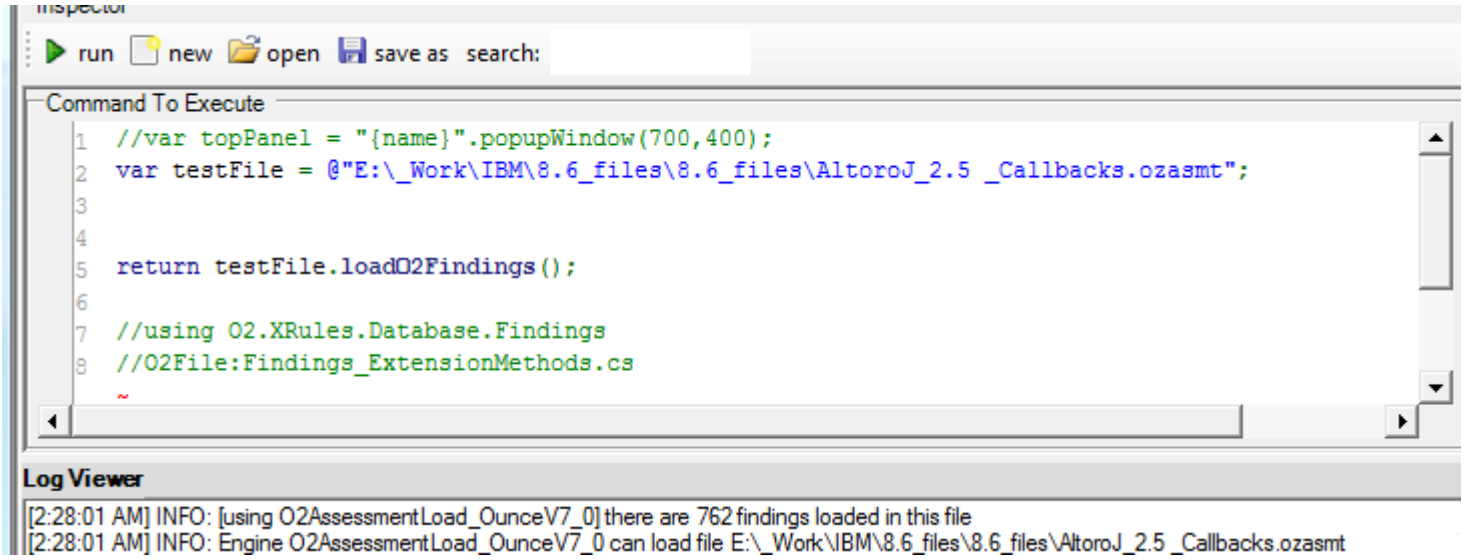
```
//var topPanel = "{name}".popupWindow(700,400);
var testFile = @"E:\_Work\IBM\8.6_files\8.6_files\AltoroJ_2.5 _Callbacks.ozasmt";


return testFile.loadO2Findings();

//using O2.XRules.Database.Findings
//O2File:Findings_ExtensionMethods.cs
```



each finding object has quite a log of info:



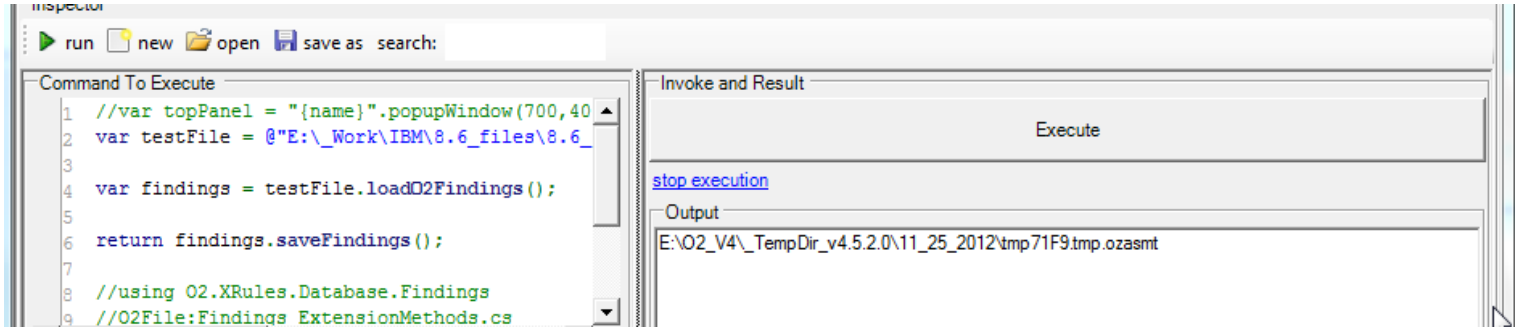
Note how the assessment was loaded using the 7.0 schema



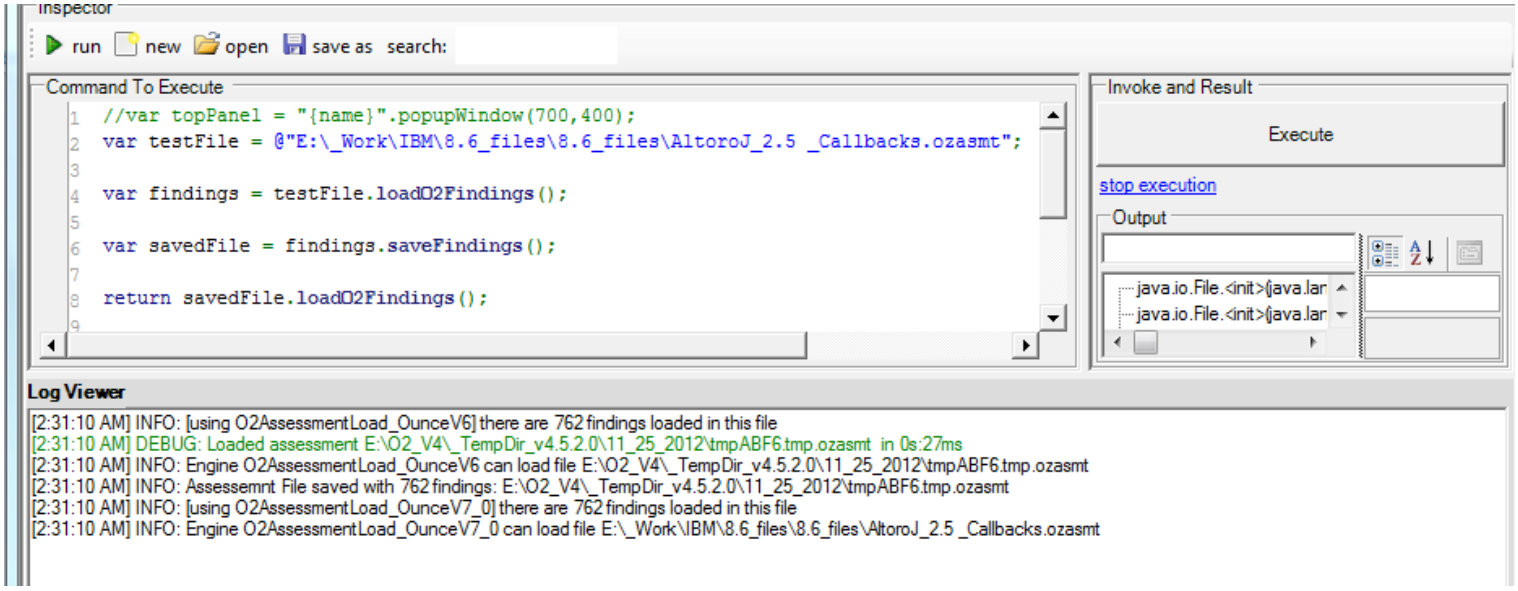
we can save the findings back like this:

```
var findings = testFile.loadO2Findings();
```

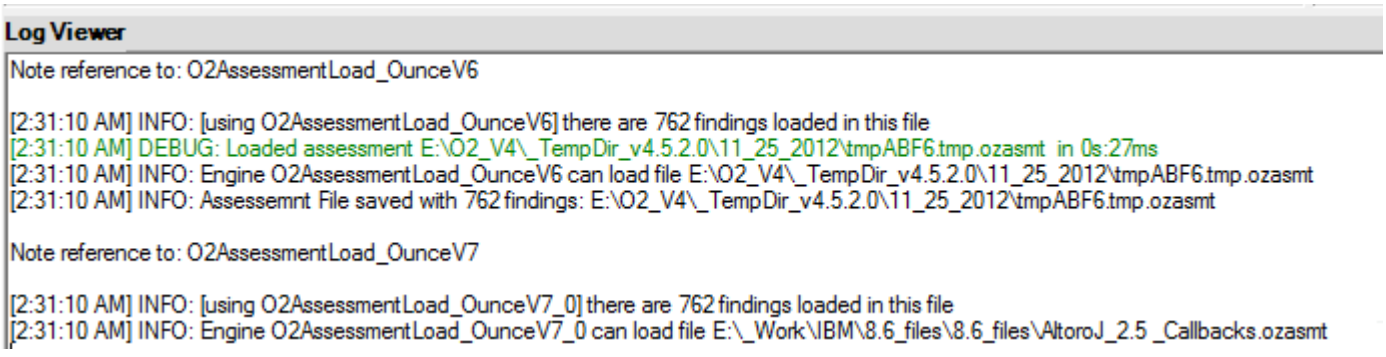
```
return findings.saveFindings();
```



If we load the saved file, we will see that it was saved with 6.0 schema:

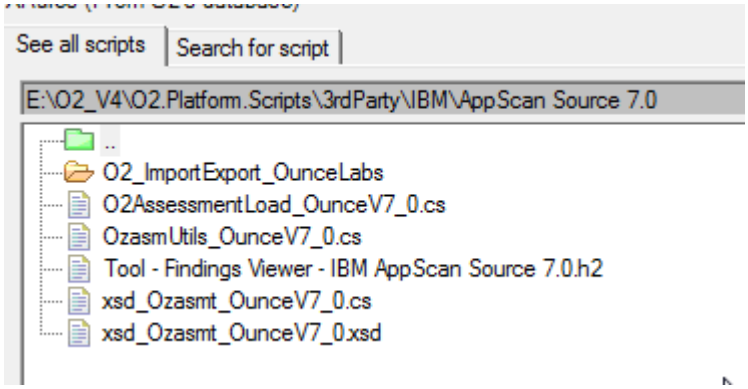


this might make it easier to read the log messages:



So the objective is to allow the round trip of load->save assessments from 8.x into 8.x (the 8.x is the same format as 7.x)

if you look in the *O2.Platform.Scripts\3rdParty\IBM\AppScan Source 7.0* folder:



you will see the xsd (Created from a bunch of ozasmt files)

xsd_Ozasmt_OunceV7_0.xsd

search config: execute on engine:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/200
  <xs:element name="AssessmentRun">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="AssessmentStats">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="VulnType">
                <xs:complexType>
                  <xs:attribute name="name" type="xs:string" use="required" />
                  <xs:attribute name="total_high_finding" type="xs:unsignedInt" use="required" />
                  <xs:attribute name="total_low_finding" type="xs:unsignedInt" use="required" />
                  <xs:attribute name="total_med_finding" type="xs:unsignedInt" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="class_total" type="xs:unsignedInt" use="required" />
            <xs:attribute name="date" type="xs:unsignedInt" use="required" />
            <xs:attribute name="duration" type="xs:unsignedInt" use="required" />
            <xs:attribute name="error_status" type="xs:unsignedInt" use="required" />
            <xs:attribute name="language_type" type="xs:unsignedInt" use="required" />
            <xs:attribute name="line_total" type="xs:unsignedInt" use="required" />
            <xs:attribute name="max_vdensity" type="xs:decimal" use="required" />
            <xs:attribute name="max_vkloc" type="xs:decimal" use="required" />
            <xs:attribute name="method_total" type="xs:unsignedInt" use="required" />
            <xs:attribute name="total_call_sites" type="xs:unsignedInt" use="required" />
            <xs:attribute name="total_call_sites_informational" type="xs:unsignedInt" use="required" />
            <xs:attribute name="total_call_sites_not_vulnerable" type="xs:unsignedInt" use="required" />
```

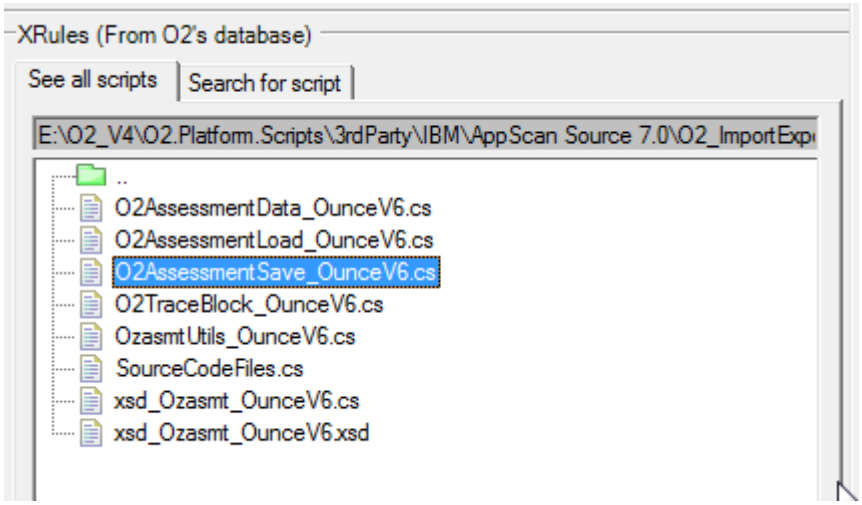
and the respective C# file for that xsd:

xsd_Ozasmt_OunceV7_0.xsd | xsd_Ozasmt_OunceV7_0.cs

search | config: | compile code: | execute code:

```
// This file is part of the OWASP O2 Platform (http://www.owasp.org/index.php/OWASP_O2_Platform) and is relea
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated by a tool.
//     Runtime Version:2.0.50727.3053
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

//
// This source code was auto-generated by xsd, Version=2.0.50727.1432.
//
namespace O2.XRules.ThirdPary.IBM
{
    /// <remarks/>
    [System.CodeDom.Compiler.GeneratedCodeAttribute("xsd", "2.0.50727.1432")]
    [System.SerializableAttribute()]
    [System.Diagnostics.DebuggerStepThroughAttribute()]
    [System.ComponentModel.DesignerCategoryAttribute("code")]
    [System.Xml.Serialization.XmlTypeAttribute(AnonymousType=true)]
    [System.Xml.Serialization.XmlRootAttribute(Namespace="", IsNullable=false)]
    public partial class AssessmentRun {

        private AssessmentRunAssessmentStats assessmentStatsField;

        private AssessmentRunAssessmentConfig assessmentConfigField;

        private AssessmentRunSharedDataStats sharedDataStatsField;

        private AssessmentRunString[] stringPoolField;

        private AssessmentRunFile[] filePoolField;

        private AssessmentRunSite[] sitePoolField;
```

Another good file to take a look at is the:



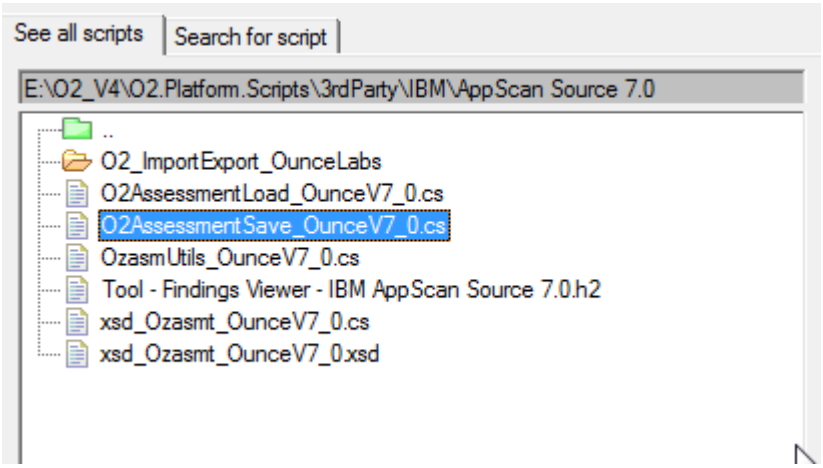
which has the save logic for the 6.0 save:

xsd_Ozasmt_OunceV7_0.xsd | xsd_Ozasmt_OunceV7_0.cs | O2AssessmentSave_OunceV6.cs

search | config: | compile code: | execute code: save(System.Collections

```
// This file is part of the OWASP O2 Platform (http://www.owasp.org/index.php/OWASP_O2_Platform) and is released u
using System;
using System.Collections.Generic;
using O2.DotNetWrappers.DotNet;
using O2.Interfaces.O2Findings;
using O2.Kernel;
//O2File:xsd_Ozasmt_OunceV6.cs
//O2File:OzasmtUtils_OunceV6.cs

namespace O2.ImportExport.OunceLabs.Ozasmt_OunceV6
{
    public class O2AssessmentSave_OunceV6 : IO2AssessmentSave
    {
        public AssessmentRun assessmentRun {get;set;}

        public O2AssessmentSave_OunceV6()
        {
            engineName = "O2AssessmentSave_OunceV6";
            assessmentRun = OzasmtUtils_OunceV6.getDefaultAssessmentRunObject();
        }

        public string engineName {get; set;}


        public string save(List<IO2Finding> o2Findings)
        {
            string tempOzasmtFile = PublicDI.config.getTempFileInTempDirectory("ozasmt");

            return (save(o2Findings,tempOzasmtFile)) ? tempOzasmtFile : "";
        }

        public bool save(List<IO2Finding> o2Findings, string sPathToSaveAssessment)
        {
            return save(assessmentRun.name, o2Findings, sPathToSaveAssessment);
        }

        public bool save(string assessmentName, IEnumerable<IO2Finding> o2Findings, string sPathToSaveAssessment)
        {
            createAssessmentRunObject(assessmentName, o2Findings);
            return OzasmtUtils_OunceV6.SaveAssessmentRun(assessmentRun, sPathToSaveAssessment);
        }
```

so lets save this file as 7_0



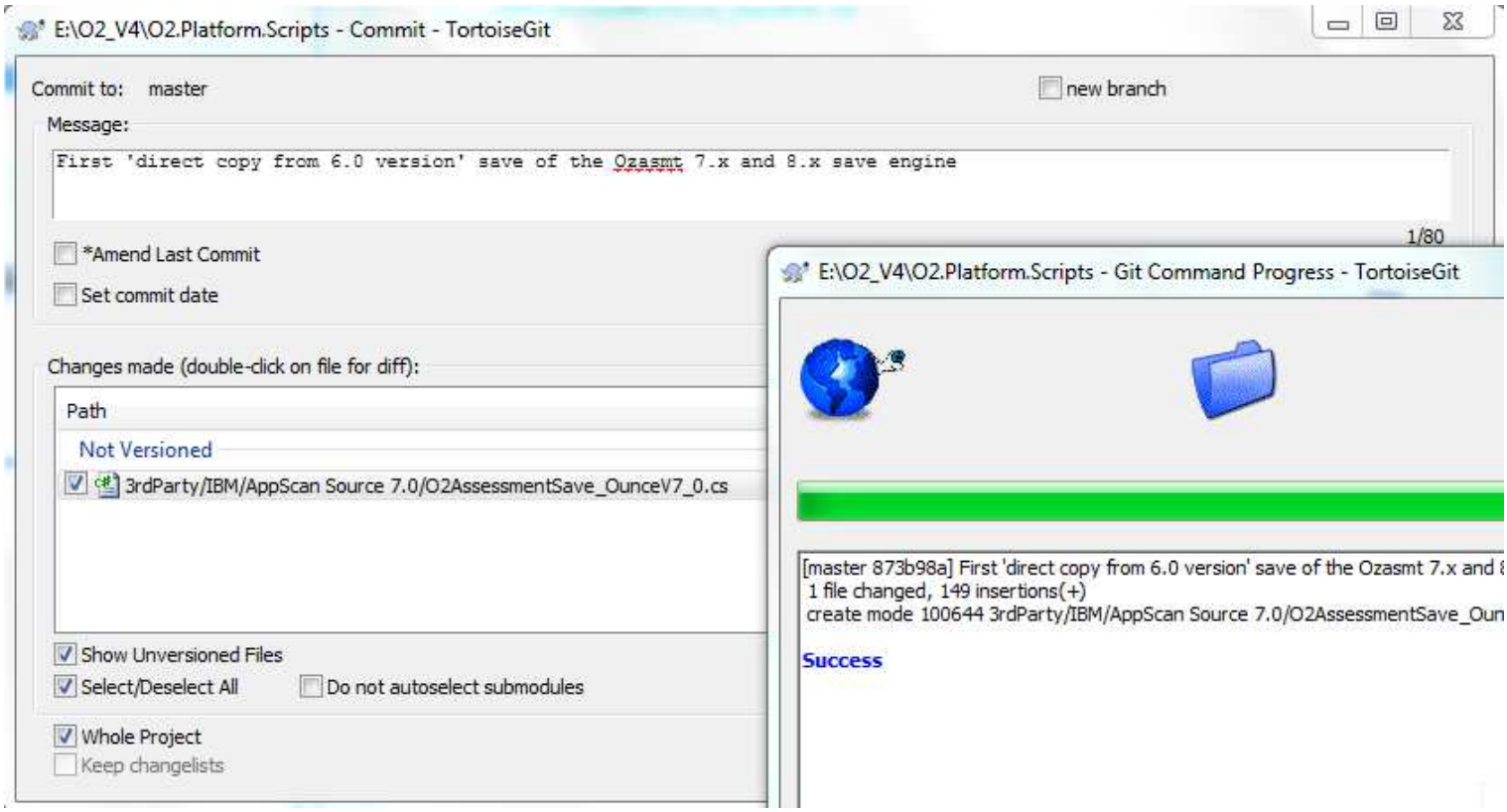
and change it enough so that it compiles

O2AssessmentSave_OunceV7_0.cs | xsd_Ozasmt_OunceV7_0.xsd | xsd_Ozasmt_OunceV7_0.cs

search | config: | compile code: | execute code: save(System.Collections

```
// This file is part of the OWASP O2 Platform (http://www.owasp.org/index.php/OWASP_O2_Platform) and is released under
using System;
using System.Collections.Generic;
using O2.DotNetWrappers.DotNet;
using O2.Interfaces.O2Findings;
using O2.Kernel;
using O2.XRules.ThirdPary.IBM;
//O2File:xsd_Ozasmt_OunceV7_0.cs
//___O2File:OzasmtUtils_OunceV6.cs

namespace O2.ImportExport.OunceLabs.Ozasmt_OunceV6
{
    public class O2AssessmentSave_OunceV7 : IO2AssessmentSave
    {
        public AssessmentRun assessmentRun {get;set;}

        public O2AssessmentSave_OunceV7()
        {
            engineName = "O2AssessmentSave_OunceV7";
            //assessmentRun = OzasmtUtils_OunceV7.getDefaultAssessmentRunObject();
        }

        public string engineName {get; set;}


        public string save(List<IO2Finding> o2Findings)
        {
            string tempOzasmtFile = PublicDI.config.getTempFileInTempDirectory("ozasmt");

            return (save(o2Findings,tempOzasmtFile)) ? tempOzasmtFile : "";
        }

        public bool save(List<IO2Finding> o2Findings, string sPathToSaveAssessment)
        {
            return save(assessmentRun.name, o2Findings, sPathToSaveAssessment);
        }

        public bool save(string assessmentName, IEnumerable<IO2Finding> o2Findings, string sPathToSaveAssessment)
        {
            //createAssessmentRunObject(assessmentName, o2Findings);
            //return OzasmtUtils_OunceV6.SaveAssessmentRun(assessmentRun, sPathToSaveAssessment);
            return false;
        }
```

To keep a better track on the changes that are going to be made to this file, here is its commit:



back in the C# Repl script, let's load the findings (from 8x into memory) and store the findings object in an o2 cache object (this will allow us to have access to the findings on each REPL execution without needing to load the findings from disk)**:**

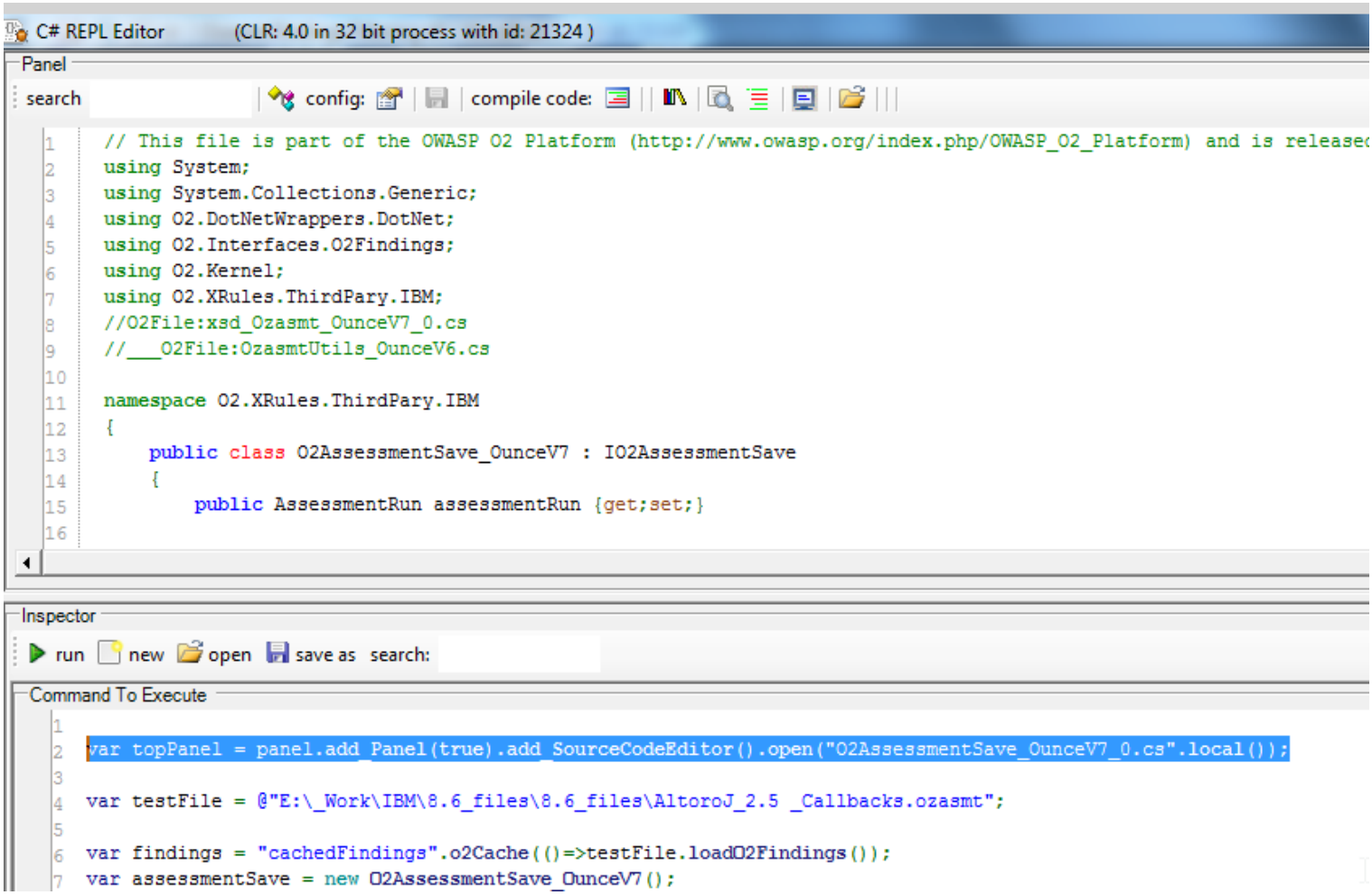
```
var testFile = @"E:\_Work\IBM\8.6_files\8.6_files\AltoroJ_2.5 _Callbacks.ozasmt";

var findings = "cachedFindings".o2Cache(()=>testFile.loadO2Findings());
return findings;
```

Next lets create an instance of the the *O2AssessmentSave_OunceV7* object

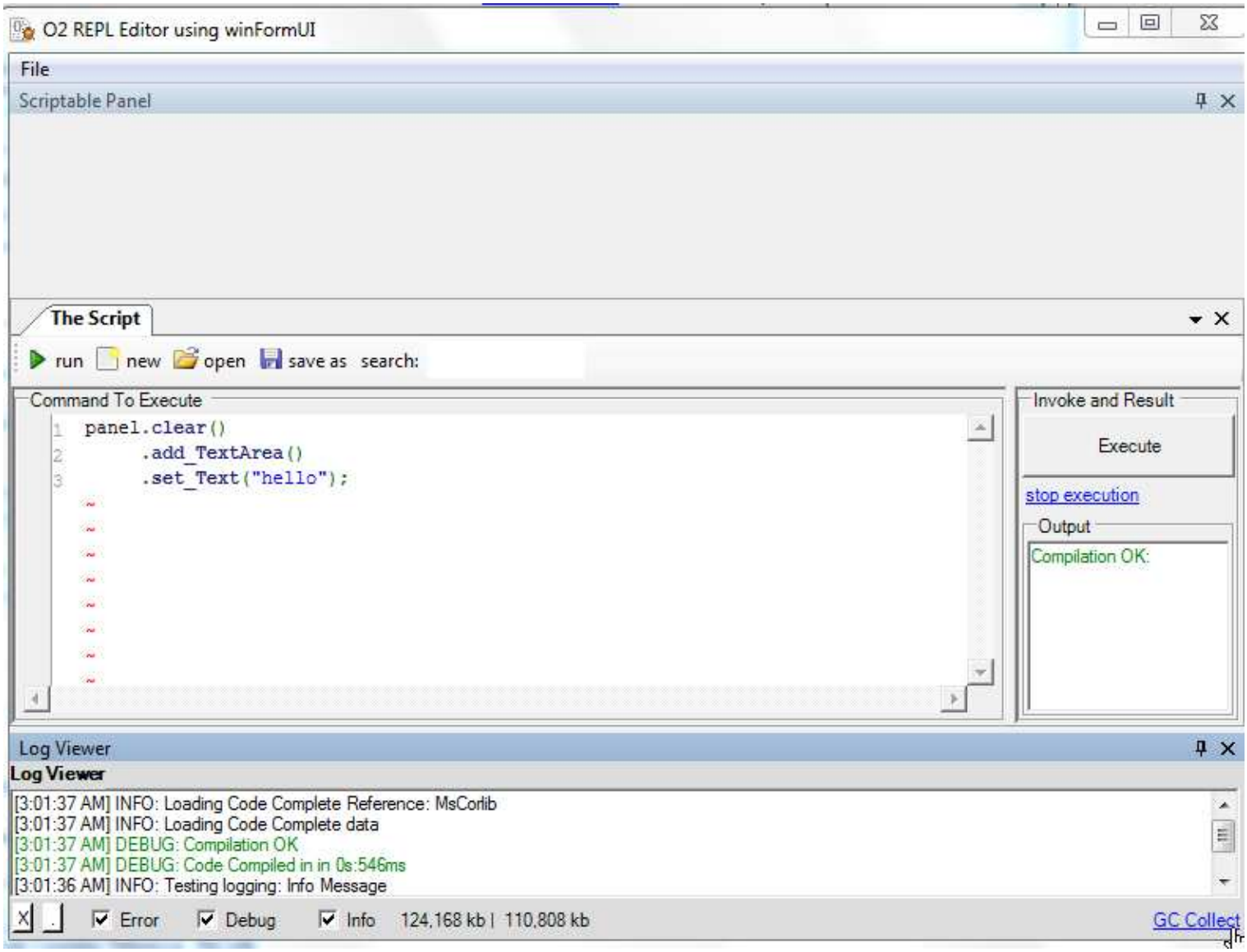
```
var testFile = @"E:\_Work\IBM\8.6_files\8.6_files\AltoroJ_2.5 _Callbacks.ozasmt";

var findings = "cachedFindings".o2Cache(()=>testFile.loadO2Findings());
var assessmentSave = new O2AssessmentSave_OunceV7();

return assessmentSave;

//using O2.XRules.ThirdPary.IBM
//using O2.XRules.Database.Findings
//O2File:O2AssessmentSave_OunceV7_0.cs
//O2File:Findings_ExtensionMethods.cs
```

To make our life easier developing this script, lets open the *O2AssessmentSave_OunceV7_0.cs* file in this C# REPL editor:

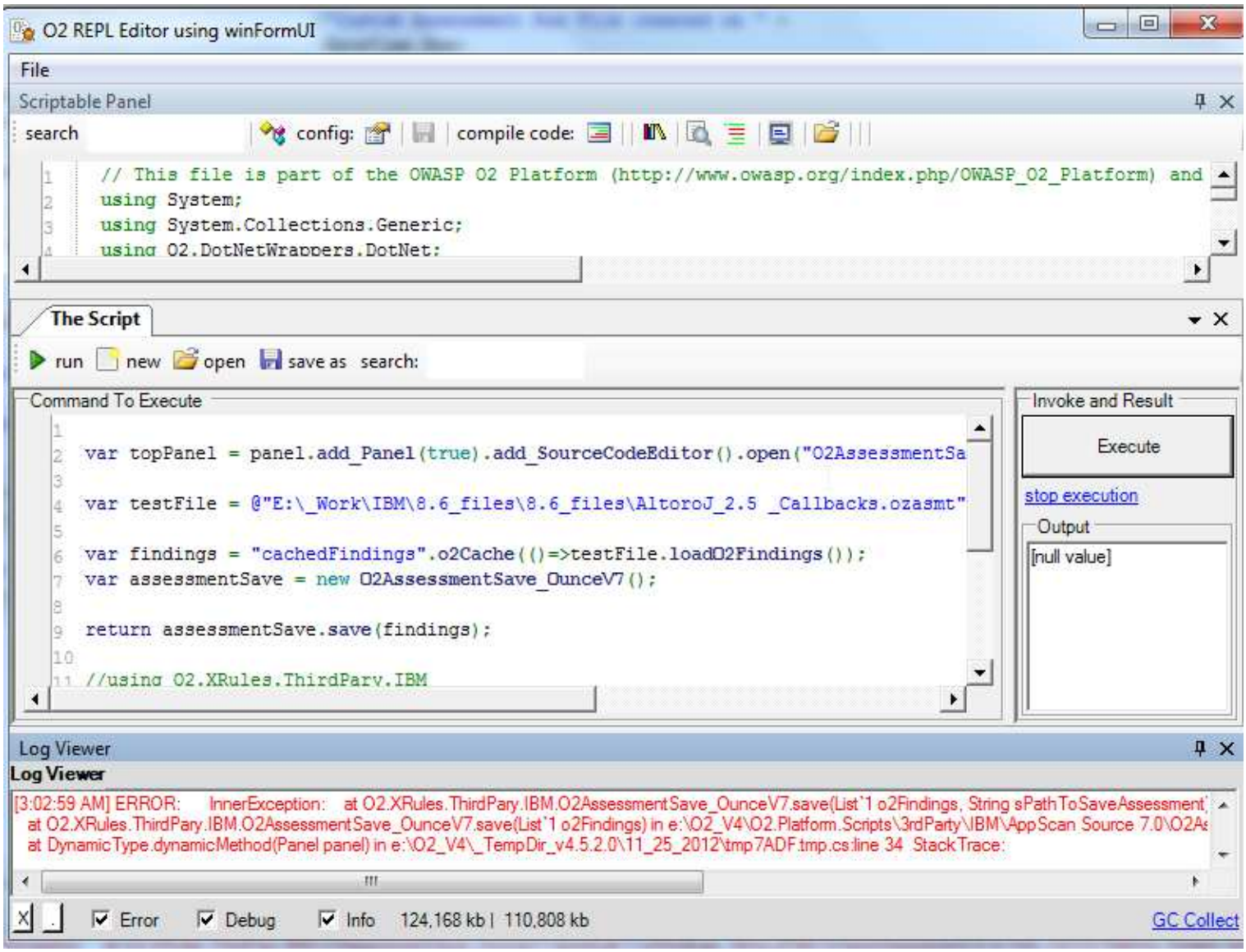
```
panel.add_Panel(true).add_SourceCodeEditor().open("O2AssessmentSave_OunceV7_0.cs".local());
```



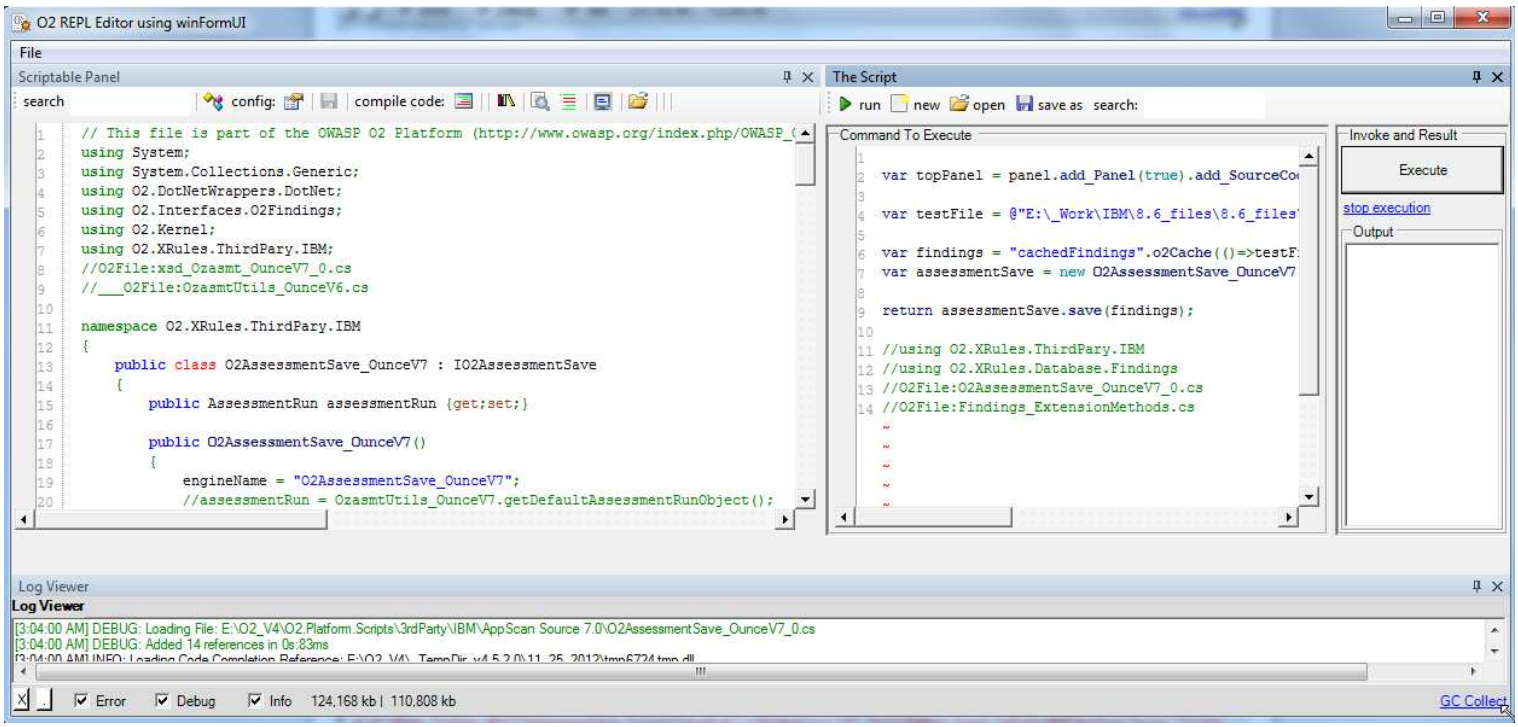
Actually the *Util - O2 C# REPL using WinFormsUI.h2* script might be more usuful



Since once we have our file loaded on the top panel:



We can put the C# file on the left and the C# REPL script on the right (with the log viewer below)



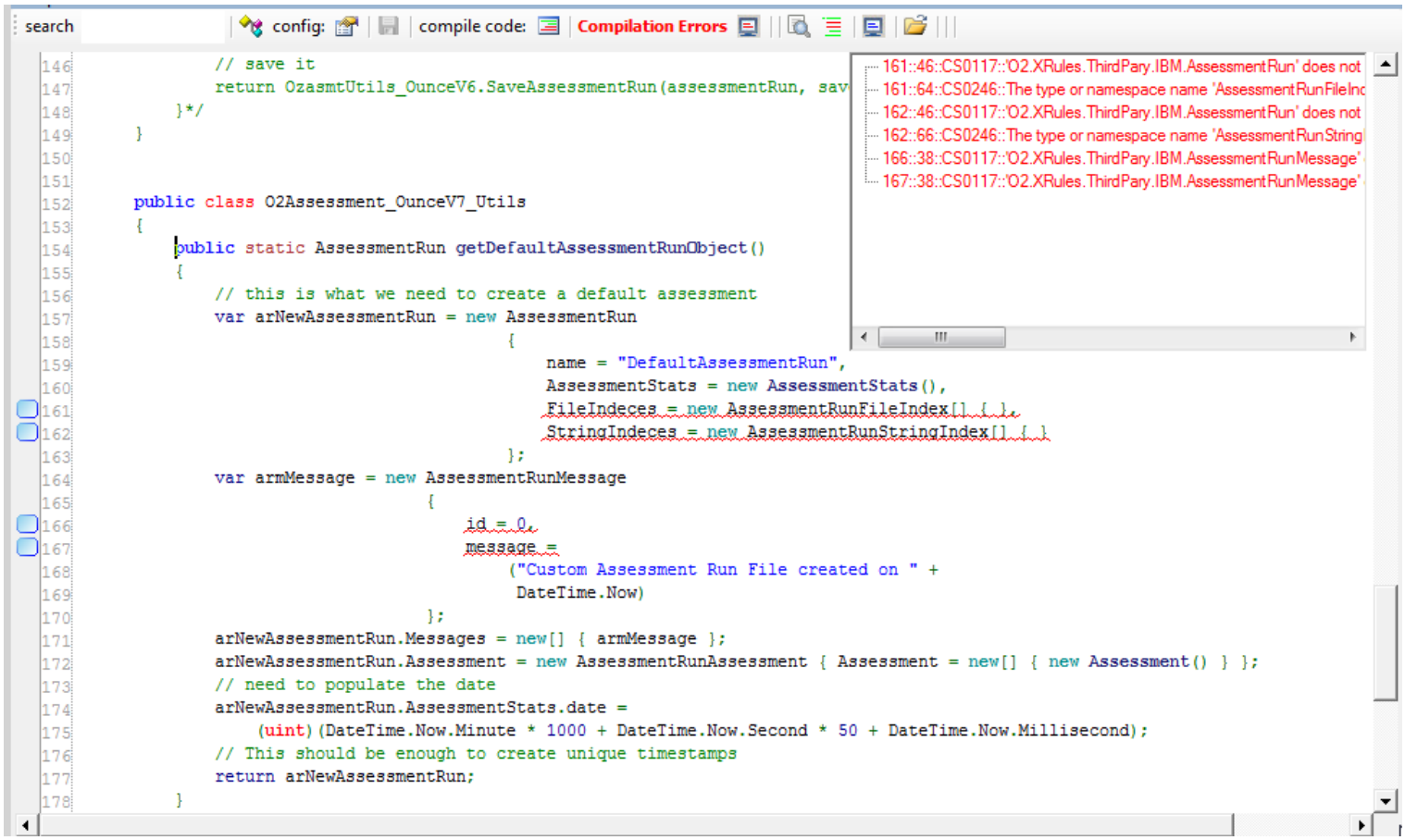
Lets create an class to hold a couple util methods we will need:

```
public class O2Assessment_OunceV7_Utils
{

}
```

Here is the *getDefaultAssessmentRunObject* method from the v6.0


```
454
455         public static AssessmentRun getDefaultAssessmentRunObject()
456         {
457             // this is what we need to create a default assessment
458             var arNewAssessmentRun = new AssessmentRun
459                                          {
460                                              name = "DefaultAssessmentRun",
461                                              AssessmentStats = new AssessmentStats(),
462                                              FileIndeces = new AssessmentRunFileIndex[] { },
463                                              StringIndeces = new AssessmentRunStringIndex[] { }
464                                          };
465             var armMessage = new AssessmentRunMessage
466                                  {
467                                      id = 0,
468                                      message =
469                                          ("Custom Assessment Run File created on " +
470                                           DateTime.Now)
471                                  };
472             arNewAssessmentRun.Messages = new[] { armMessage };
473             arNewAssessmentRun.Assessment = new AssessmentRunAssessment { Assessment = new[] { new Assessment() } };
474             // need to populate the date
475             arNewAssessmentRun.AssessmentStats.date =
476                 (uint)(DateTime.Now.Minute * 1000 + DateTime.Now.Second * 50 + DateTime.Now.Millisecond);
477             // This should be enough to create unique timestamps
478             return arNewAssessmentRun;
479         }
```

which we wil need to convert into the 7_0 version (compilation fails because some of those objects don't exist any more)

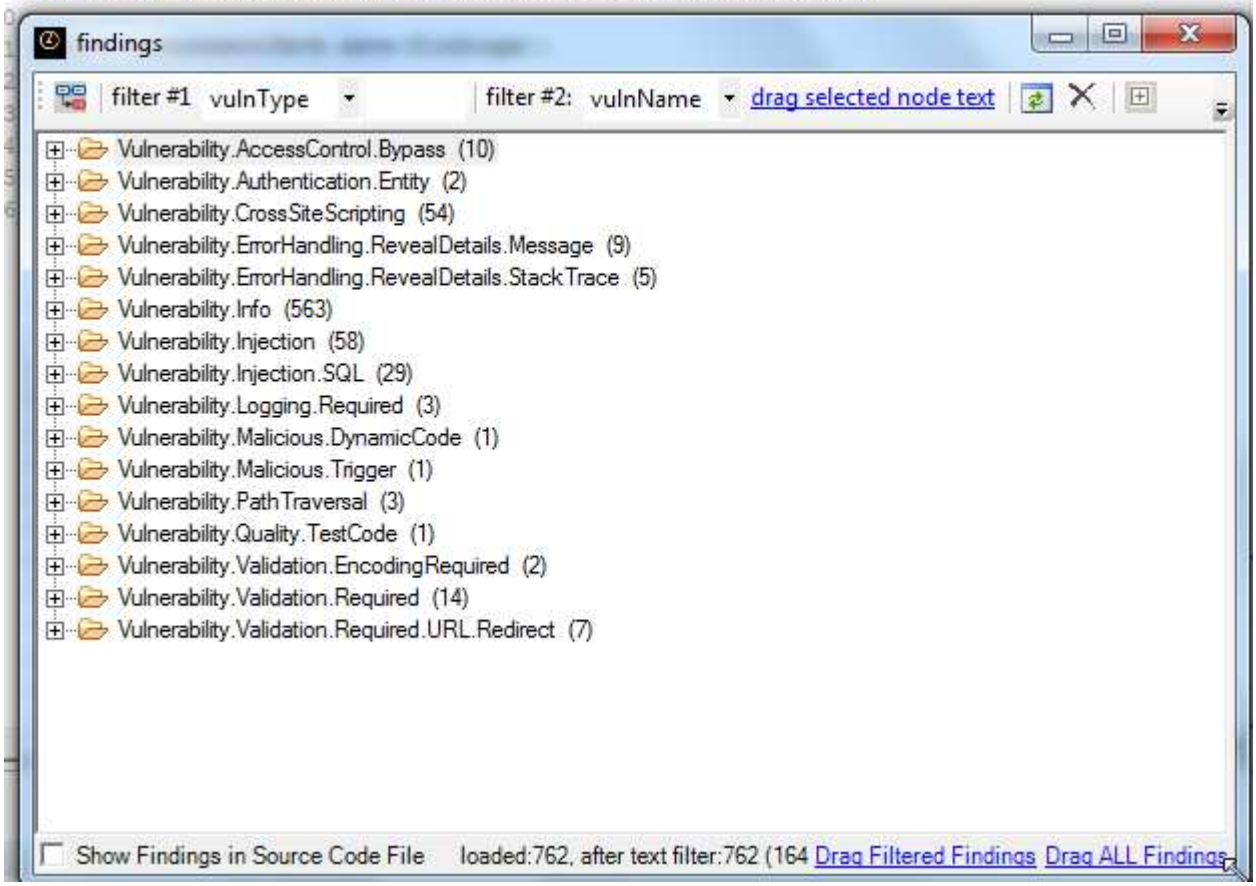


Tip: if you want to open a new findings viewer window to take a look at some findings, you can use:

```
var testFile = @"E:\_Work\IBM\8.6_files\8.6_files\AltoroJ_2.5 _Callbacks.ozasmt";
var findings = "cachedFindings".o2Cache(()=>testFile.loadO2Findings());

"findings".popupWindow().add_FindingsViewer().show(findings);
```

And a popup window will appear

```
"findings".popupWindow().add_FindingsViewer().show(findings);
```



Back to the *getDefaultAssessementRunObject*

```
151
152     public class O2Assessment_OunceV7_Utils
153     {
154         public static AssessmentRun getDefaultAssessmentRunObject()
155         {
156             // this is what we need to create a default assessment
157             var arNewAssessmentRun = new AssessmentRun
158                                          {
159                                              name = "DefaultAssessmentRun",
160 //                                             AssessmentStats = new AssessmentStats(),
161 //                                             FileIndeces = new AssessmentRunFileIndex[] { },
162 //                                             StringIndeces = new AssessmentRunStringIndex[] { }
163                                          };
164             var armMessage = new AssessmentRunMessage
165                                  {
166 //                                     id = 0,
167 //                                     message =
168 //                                         ("Custom Assessment Run File created on " +
169 //                                          DateTime.Now)
170                                  };
171             arNewAssessmentRun.Messages = new[] { armMessage };
172             arNewAssessmentRun.Assessment = new AssessmentRunAssessment { Assessment = new[] { new Assessment() } };
173             // need to populate the date
174             arNewAssessmentRun.AssessmentStats.date =
175                 (uint)(DateTime.Now.Minute * 1000 + DateTime.Now.Second * 50 + DateTime.Now.Millisecond);
176             // This should be enough to create unique timestamps
177             return arNewAssessmentRun;
178         }
179     }
```

here are 7.0 variables (from the C#, created from the xsd, created from the ozasmt xml files)

```
25      public partial class AssessmentRun
26      {
27                                                                  public AssessmentRunAssessmentStats     AssessmentStats     { get; set; }
28                                                                  public AssessmentRunAssessmentConfig    AssessmentConfig    { get; set; }
29                                                                  public AssessmentRunSharedDataStats     SharedDataStats     { get; set; }
30          [XmlArrayItemAttribute("String",      IsNullable=false)]  public AssessmentRunString[]            StringPool          { get; set; }
31          [XmlArrayItemAttribute("File",        IsNullable=false)]  public AssessmentRunFile[]              FilePool            { get; set; }
32          [XmlArrayItemAttribute("Site",        IsNullable=false)]  public AssessmentRunSite[]              SitePool            { get; set; }
33          [XmlArrayItemAttribute("Taint",       IsNullable=false)]  public AssessmentRunTaint[]             TaintPool           { get; set; }
34          [XmlArrayItemAttribute("FindingData", IsNullable=false)]  public AssessmentRunFindingData[]       FindingDataPool     { get; set; }
35                                                                  public AssessmentRunAssessment          Assessment          { get; set; }
36          [XmlArrayItemAttribute("Message",     IsNullable=false)]  public AssessmentRunMessage[]           Messages            { get; set; }
37          [XmlAttributeAttribute()]                                 public string                         name                { get; set; }
38          [XmlAttributeAttribute()]                                 public string                         version             { get; set; }
39      }
```

It might be easier to read like this:

```
public AssessmentRunAssessmentStats     AssessmentStats     { get; set; }
public AssessmentRunAssessmentConfig    AssessmentConfig    { get; set; }
public AssessmentRunSharedDataStats     SharedDataStats     { get; set; }
public AssessmentRunString[]            StringPool          { get; set; }
public AssessmentRunFile[]              FilePool            { get; set; }
public AssessmentRunSite[]              SitePool            { get; set; }
public AssessmentRunTaint[]             TaintPool           { get; set; }
public AssessmentRunFindingData[]       FindingDataPool     { get; set; }
public AssessmentRunAssessment          Assessment          { get; set; }
public AssessmentRunMessage[]           Messages            { get; set; }
public string                           name                { get; set; }
public string                           version             { get; set; }
```

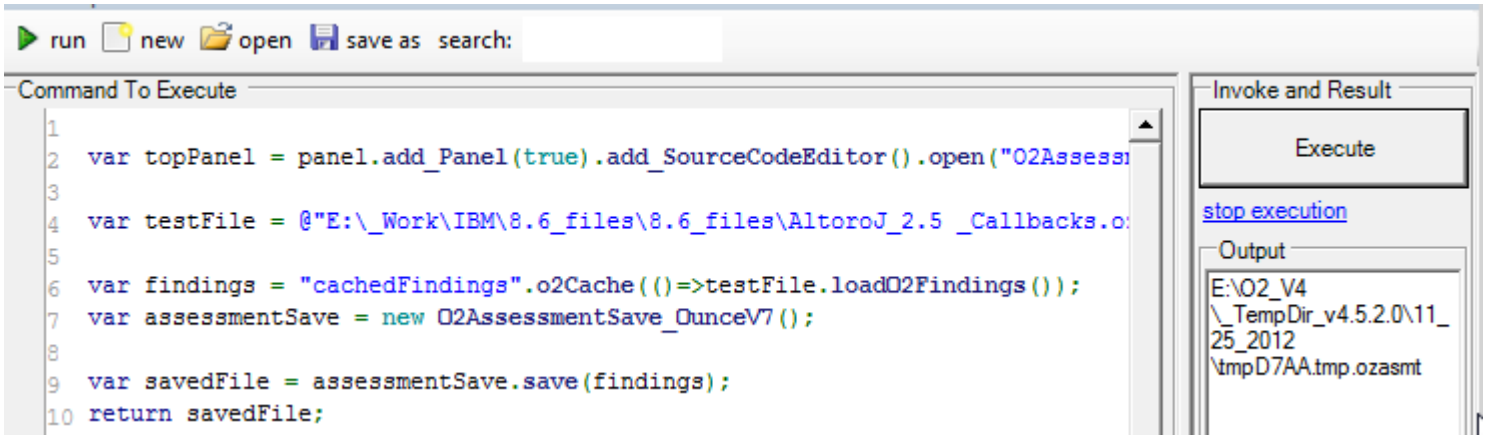
here is how to create these variables:

```
154         public static AssessmentRun getDefaultAssessmentRunObject()
155         {
156             // this is what we need to create a default assessment
157             var defaultName = "DefaultAssessmentRun_v8";
158             var defaultVersion  = "8.6.0.0";
159             var AssessmentStats = new AssessmentRunAssessmentStats();
160             var AssessmentConfig = new AssessmentRunAssessmentConfig();
161             var SharedDataStats = new AssessmentRunSharedDataStats();
162             var StringPool = new AssessmentRunString[] {};
163             var FilePool = new AssessmentRunFile[] {};
164             var SitePool = new AssessmentRunSite[] {};
165             var TaintPool = new AssessmentRunTaint[] {};
166             var FindingDataPool = new AssessmentRunFindingData[] {};
167             var Assessment = new AssessmentRunAssessment();
168             var Messages = new AssessmentRunMessage[]  {};
169             var name = defaultName;
170             var version = defaultVersion;
171
172             var arNewAssessmentRun = new AssessmentRun
173                                         {
174                                             name = "DefaultAssessmentRun",
175 //                                            AssessmentStats = new AssessmentStats(),
176 //                                            FileIndeces = new AssessmentRunFileIndex[] { },
177 //                                            StringIndeces = new AssessmentRunStringIndex[] { }
178                                         };
```

and now as part of the AssessmentRun constructor:

```
        public static AssessmentRun getDefaultAssessmentRunObject()
        {
            // this is what we need to create a default assessment
            var defaultName = "DefaultAssessmentRun_v8";
            var defaultVersion  = "8.6.0.0";

            var arNewAssessmentRun = new AssessmentRun
                                        {
                                            AssessmentStats = new AssessmentRunAssessmentStats(),
                                            AssessmentConfig = new AssessmentRunAssessmentConfig(),
                                            SharedDataStats = new AssessmentRunSharedDataStats(),
                                            StringPool = new AssessmentRunString[] {},
                                            FilePool = new AssessmentRunFile[] {},
                                            SitePool = new AssessmentRunSite[] {},
                                            TaintPool = new AssessmentRunTaint[] {},
                                            FindingDataPool = new AssessmentRunFindingData[] {},
                                            Assessment = new AssessmentRunAssessment(),
                                            Messages = new AssessmentRunMessage[] {},
                                            name = defaultName,
                                            version = defaultVersion
                                        };
```

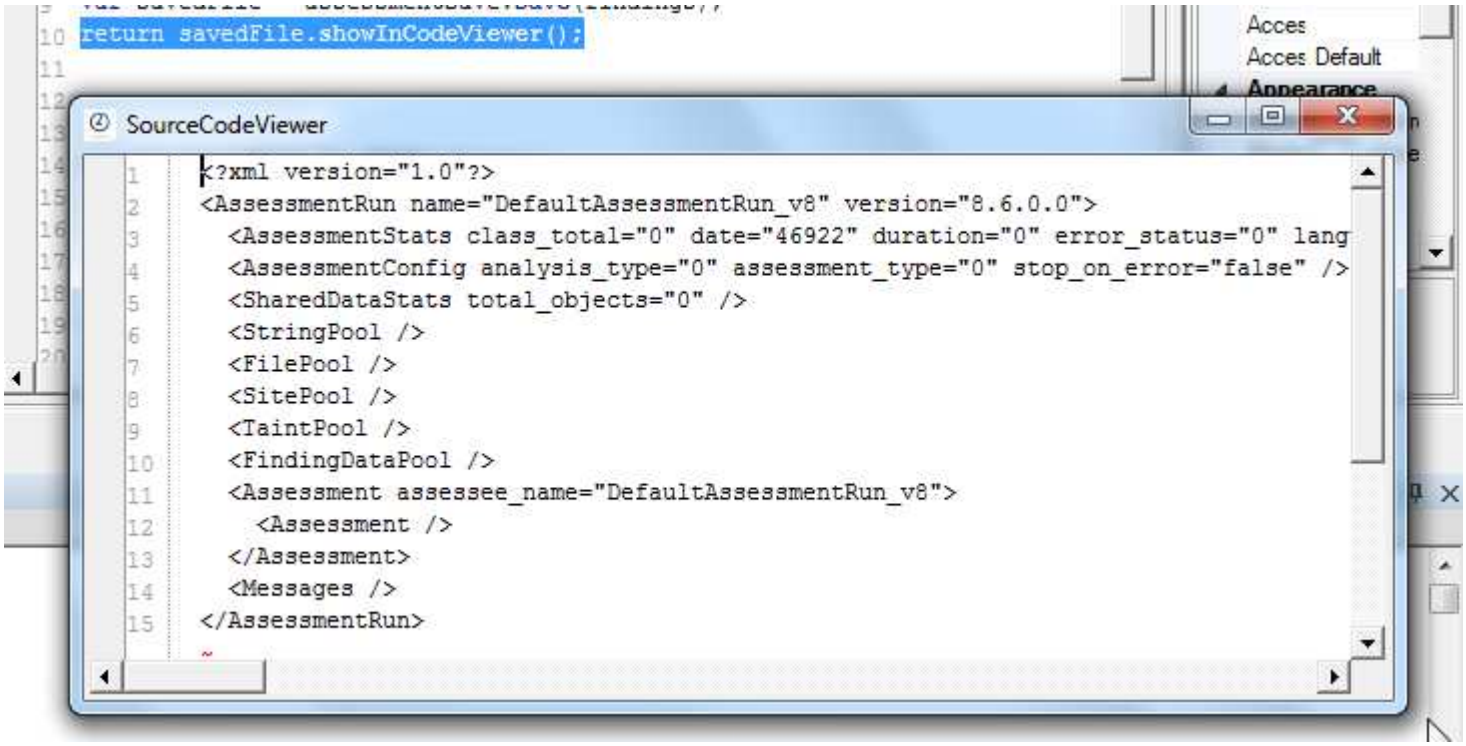
With these changes (and a couple minor tweaks) we can now create a saved 7.x (and 8.x) empty assessment file:



using

```
return savedFile.showInCodeViewer();
```

we can see what it looks like :



And this means that we now have a round trip (load - save - load) all using the 7.x load and save engines:

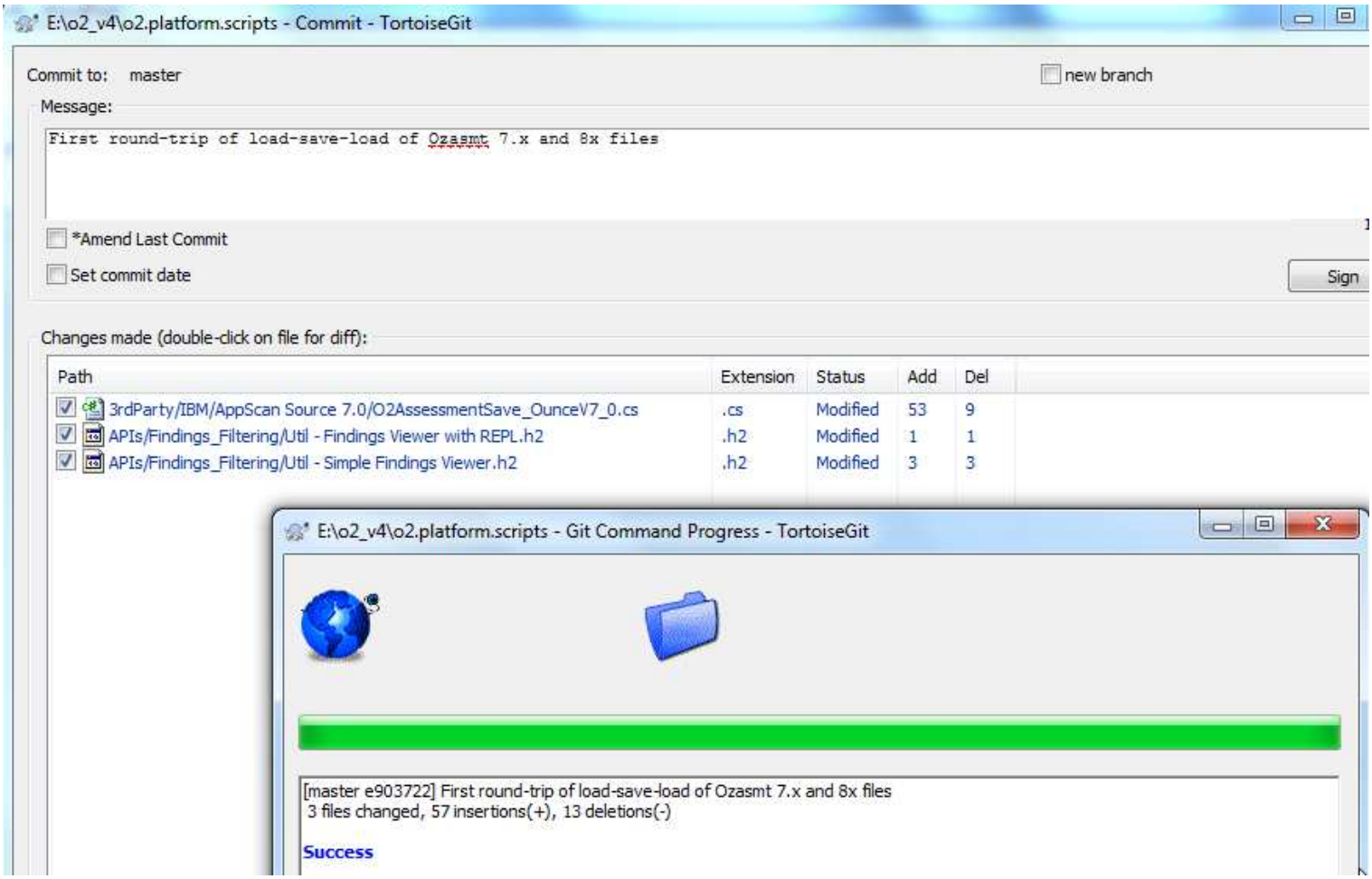
```
var testFile = @"E:\_Work\IBM\8.6_files\8.6_files\AltoroJ_2.5 _Callbacks.ozasmt";

var findings = "cachedFindings".o2Cache(()=>testFile.loadO2Findings());
var assessmentSave = new O2AssessmentSave_OunceV7();

var savedFile = assessmentSave.save(findings);

return savedFile.loadO2Findings();
```

Will show these log messages:

[3:47:23 AM] INFO: [There we no findings loaded from file: E:\O2_V4_TempDir_v4.5.2.0\11_25_2012\tmp25D9.tmp.ozasmt
[3:47:23 AM] INFO: Engine O2AssessmentLoad_OunceV7_0 can load file E:\O2_V4_TempDir_v4.5.2.0\11_25_2012\tmp25D9.tmp.ozasmt
[3:47:17 AM] DEBUG: Added 14 references in 0s:0ms

which is exactly what we wanted :)

For reference here is the commit with the changes so far:

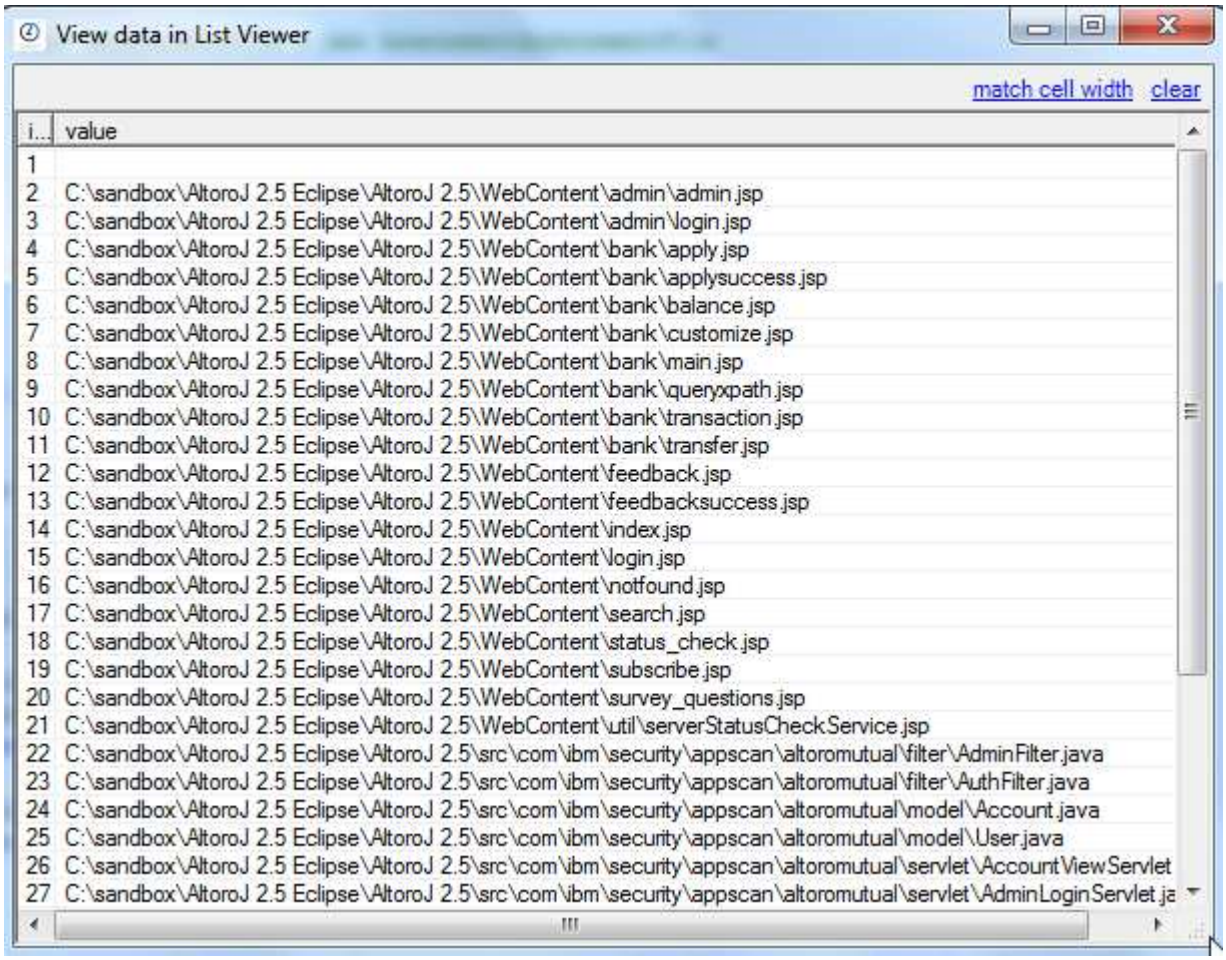


Next lets calculate the FilePool

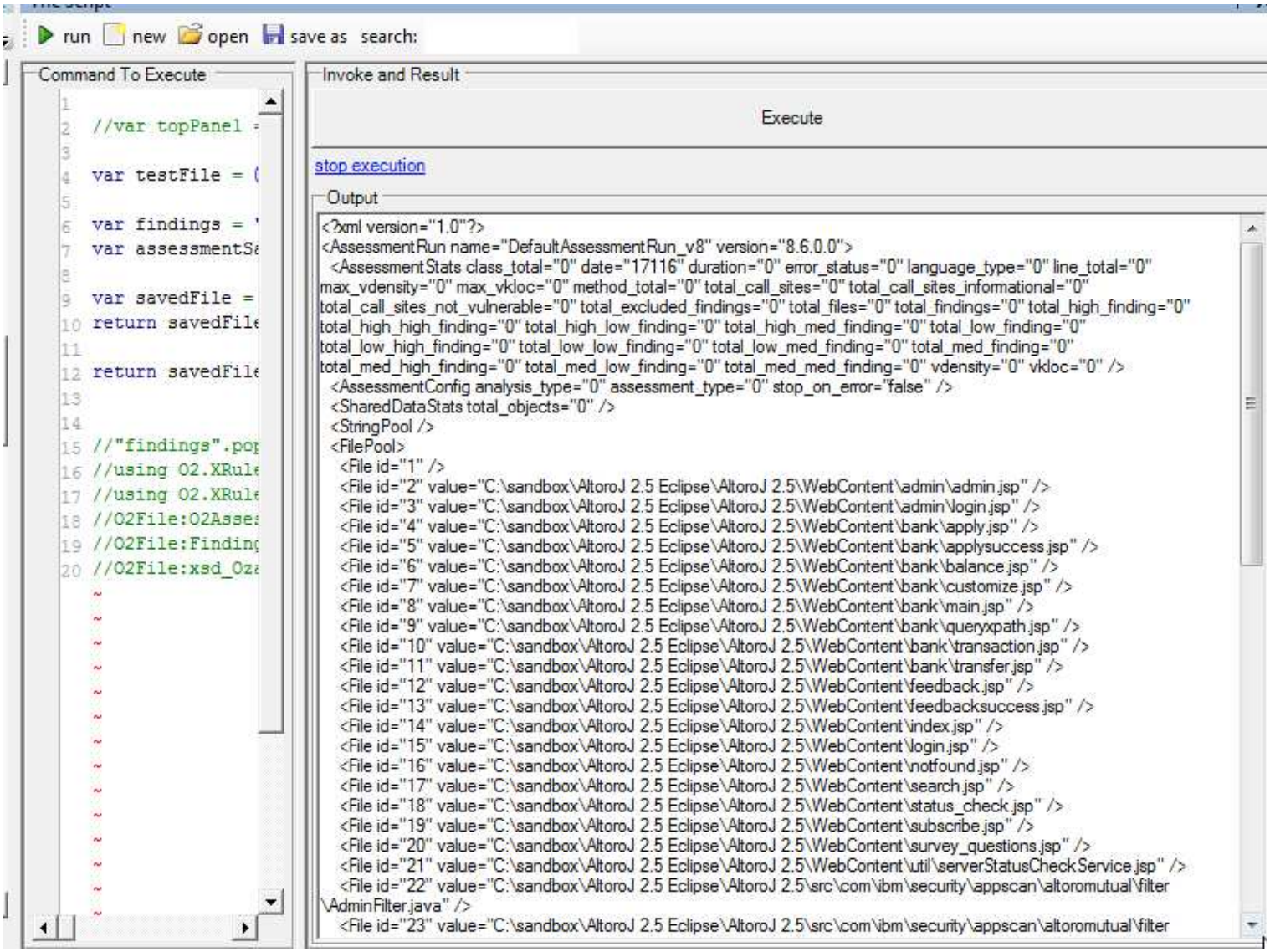
```
        public AssessmentRunFile[] getFilePool(IEnumerable<IO2Finding> o2Findings)
    {

        var uniqueFiles = (from o2Finding in o2Findings
                                   where o2Finding.file != null
                                   select o2Finding.file).distinct();

                var filePool = new List<AssessmentRunFile>();

                filePool.add(new AssessmentRunFile() { id=1 });
                UInt32 id = 2;
                foreach(var uniqueFile in uniqueFiles)
                       filePool.add(new AssessmentRunFile() { id = id++ , value = uniqueFile});
                filePool.show_In_ListView().makeColumnWidthMatchCellWidth();
                return filePool.ToArray();

    }
```

which we can quickly preview, due to the temp line (in the code above)

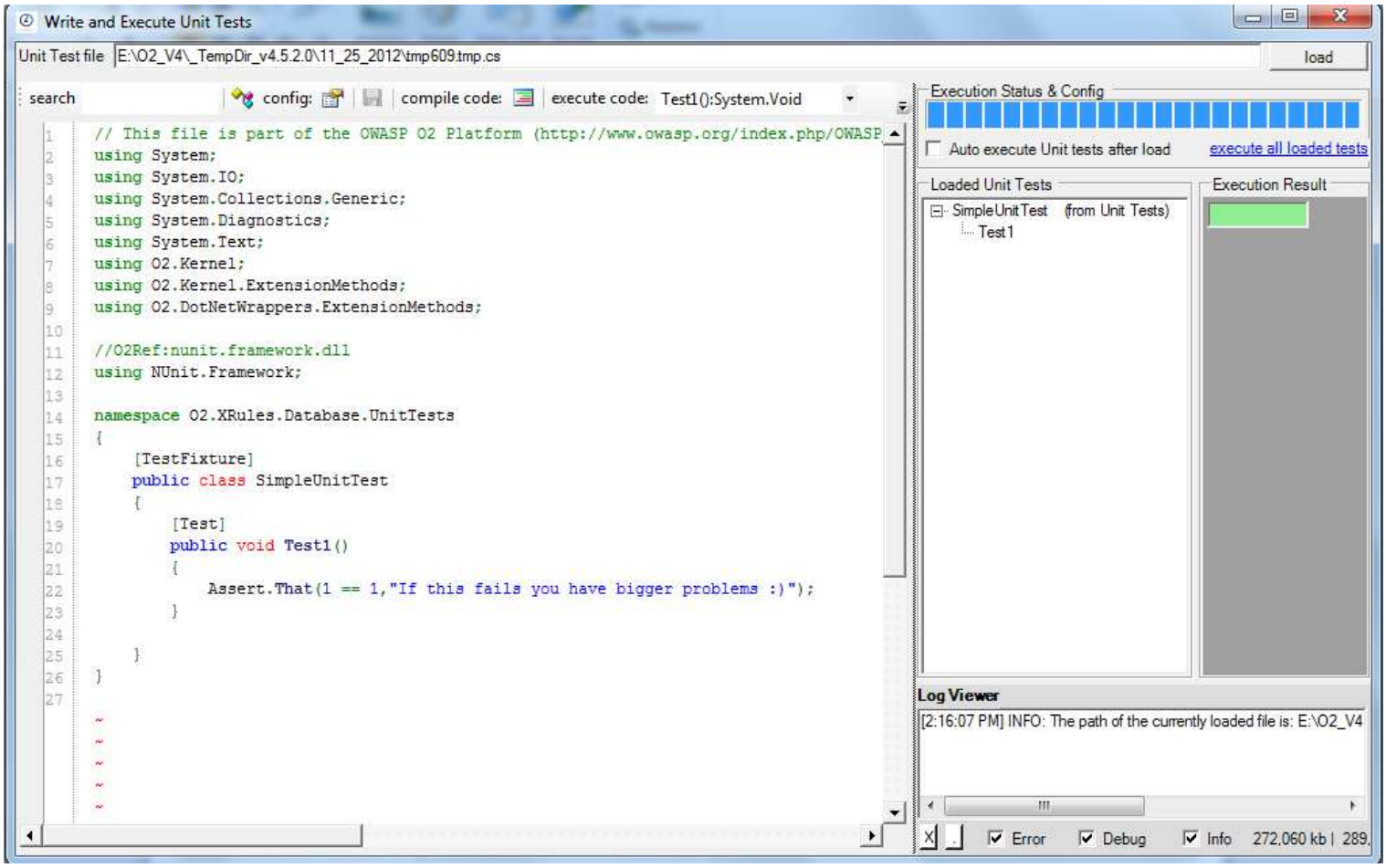
```
                filePool.show_In_ListView().makeColumnWidthMatchCellWidth();
```



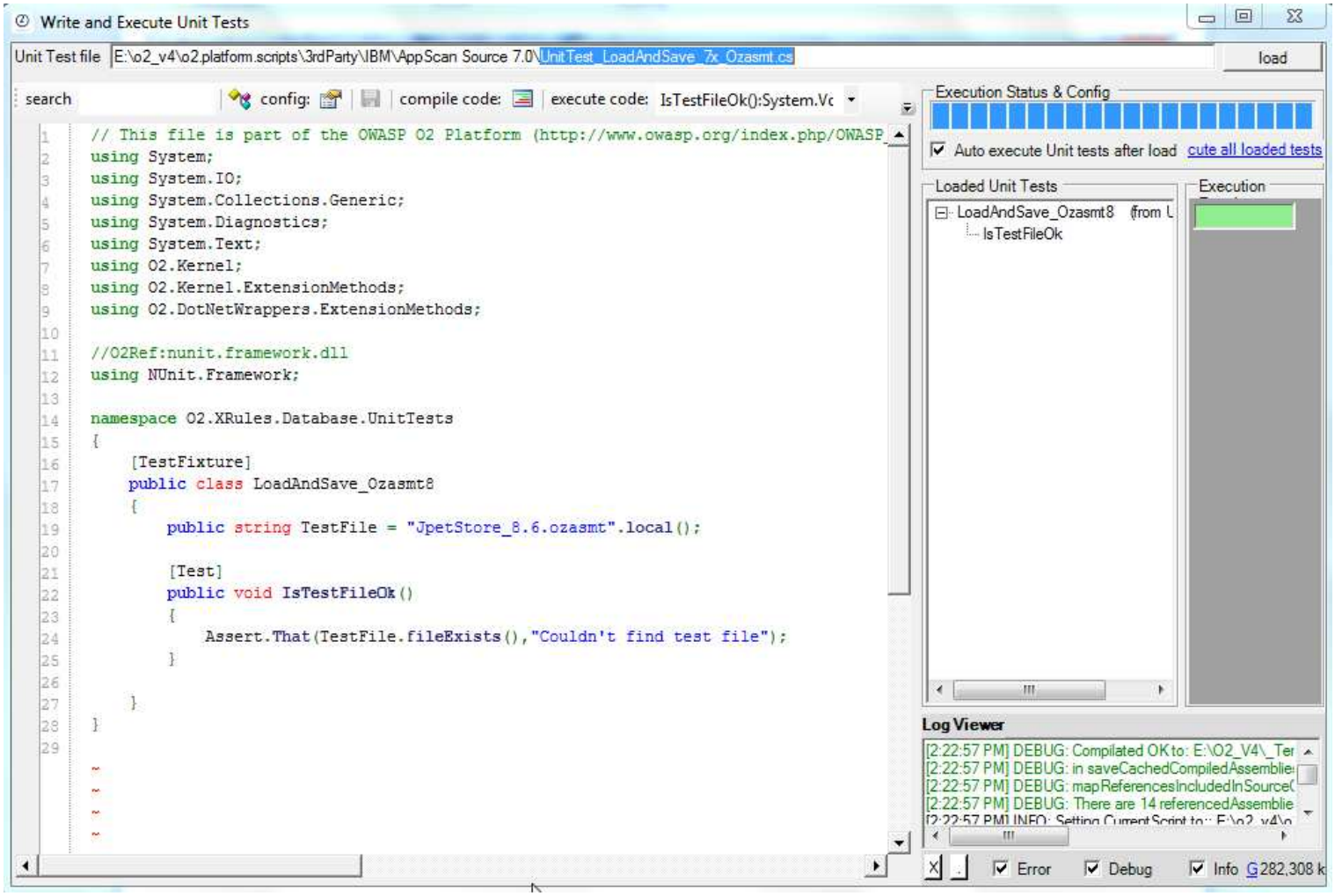
a quick look at the saved file shows that the file pool values are in there:



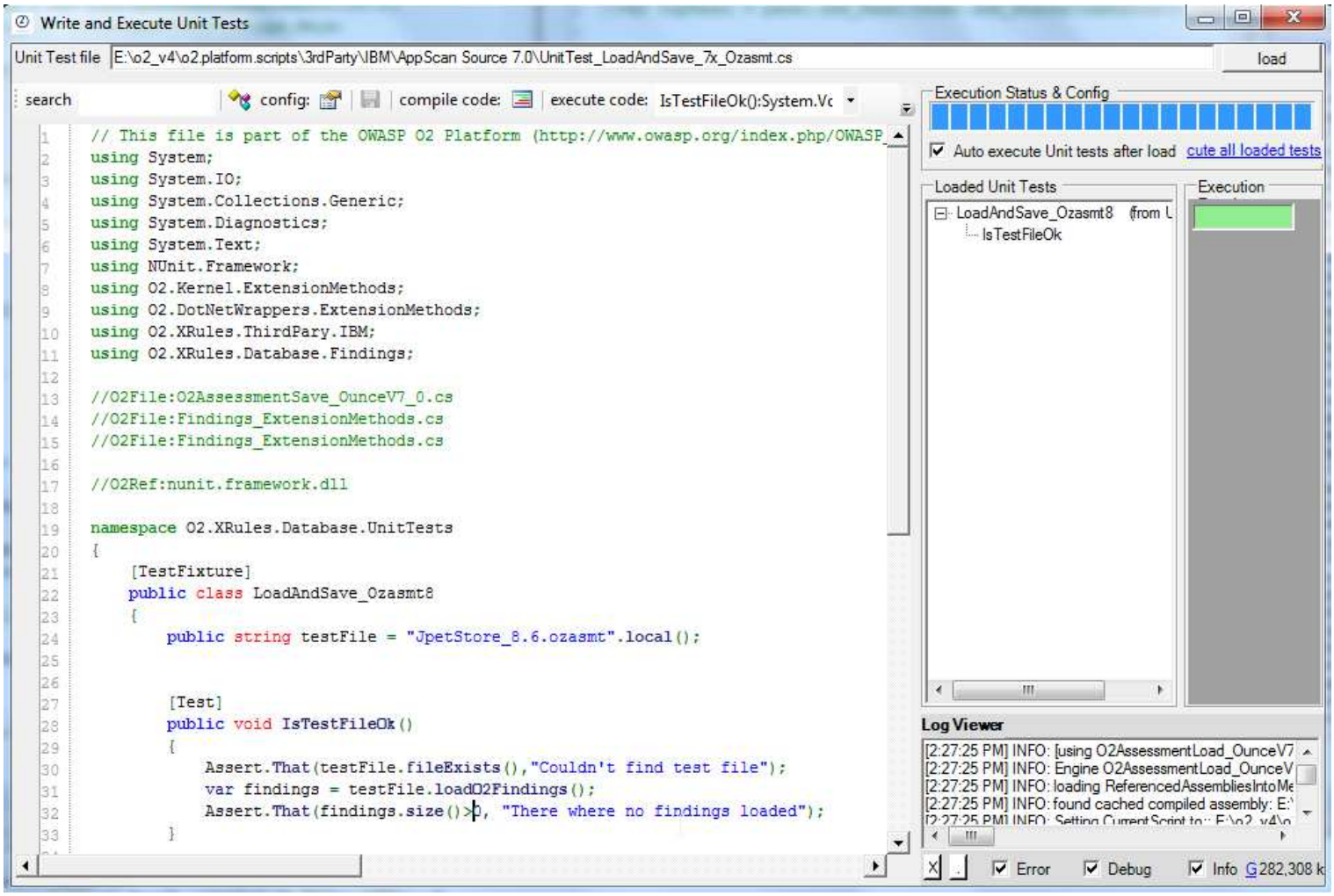
To help with the test of the conversion let's use the *Write and Execute Unit Tests.h2* script to write a UnitTest



Let's call the file *UnitTest_LoadAndSave_7x_Ozasmt.cs* and load a test ozasmt file (added to the *o2.platform.scripts _DataFiles\O2Findings* folder)



Adding the file import references, using namespaces and a check to see if there are loaded O2 findings



Moving the findings file load to the constructor

```
// This file is part of the OWASP O2 Platform (http://www.owasp.org/index.php/OWASP_O2_Platform) and is
released under the Apache 2.0 License (http://www.apache.org/licenses/LICENSE-2.0)
using System;
using System.IO;
using System.Collections.Generic;
using System.Diagnostics;
using System.Text;
using NUnit.Framework;
using O2.Interfaces.O2Findings;
using O2.DotNetWrappers.ExtensionMethods;
using O2.XRules.ThirdPary.IBM;
using O2.XRules.Database.Findings;

//O2File:O2AssessmentSave_OunceV7_0.cs
//O2File:Findings_ExtensionMethods.cs
//O2File:Findings_ExtensionMethods.cs

//O2Ref:nunit.framework.dll

namespace O2.XRules.Database.UnitTests
{
	[TestFixture]
    public class LoadAndSave_Ozasmt8
    {
       public string                    TestFile            { get; set; }
       public List<IO2Finding> Findings_Load   { get; set; }

       public LoadAndSave_Ozasmt8()
       {
              TestFile = "JpetStore_8.6.ozasmt".local();
              Findings_Load = TestFile.loadO2Findings();
       }
```

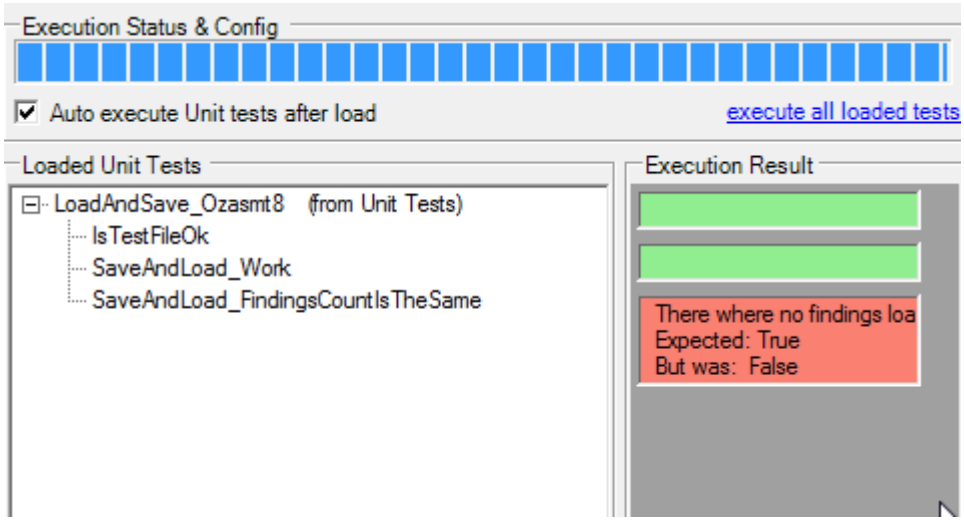
```
        [Test]
        public void IsTestFileOk()
        {
                Assert.That(TestFile.fileExists(),"Couldn't find test file");

                Assert.That(Findings_Load.size()>0, "There where no findings loaded");
        }

                [Test]
        public void SavedFileOk()
        {

        }
    }
}
```

Test to check the 7x save and load (note how the 6_1 and 6_0 load is expected to fail)

```
        [Test]
        public void SaveAndLoadWork()
        {
                // save findings using new O2AssessmentSave_OunceV7 engine
                var savedFile = new O2AssessmentSave_OunceV7().save(Findings_Load);

                // check that it exists
                Assert.That(TestFile.fileExists(),"Couldn't find O2AssessmentSave_OunceV7 saved file");

                // check that we can load the saved file with 7x
                var o2Assessment = new O2AssessmentLoad_OunceV7_0().loadFile(savedFile);
                Assert.That(o2Assessment.notNull(), "O2AssessmentLoad_OunceV7_0 failed to load");


                // check that we CAN'T load the saved file with 6.1
                o2Assessment = new O2AssessmentLoad_OunceV6_1().loadFile(savedFile);
                Assert.That(o2Assessment.isNull(), "O2AssessmentLoad_OunceV6_1 failed to load");


                // check that we CAN'T load the saved file with 6.0
                o2Assessment = new O2AssessmentLoad_OunceV6().loadFile(savedFile);
                Assert.That(o2Assessment.isNull(), "O2AssessmentLoad_OunceV6_0 failed to load");

        }
```

Test that checks if the number of findings match (this test fails at the moment)

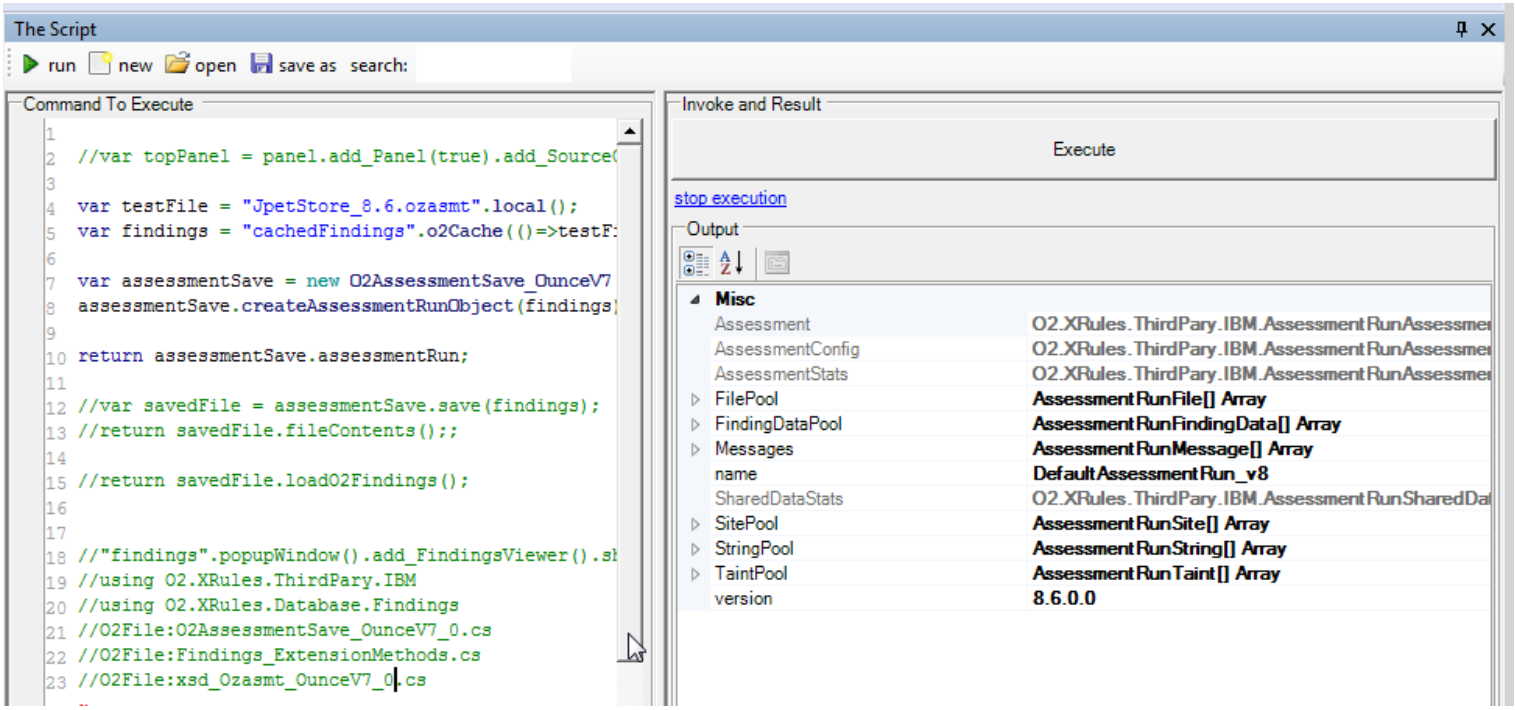
```
        [Test]
        public void SaveAndLoad_FindingsCountIsTheSame()
        {
                var savedFile = new O2AssessmentSave_OunceV7().save(Findings_Load);
                var savedFindings = savedFile.loadO2Findings();
                Assert.That(savedFindings.size() > 0 , "There where no findings loaded");
                Assert.That(savedFindings.size() == Findings_Load.size() , "Finding's size doesn't match");
        }
```



Back in the REPL (which is an easier/faster place to write some types of Unit Tests), here is how to invoke the save engine directly (without needing to save the file)

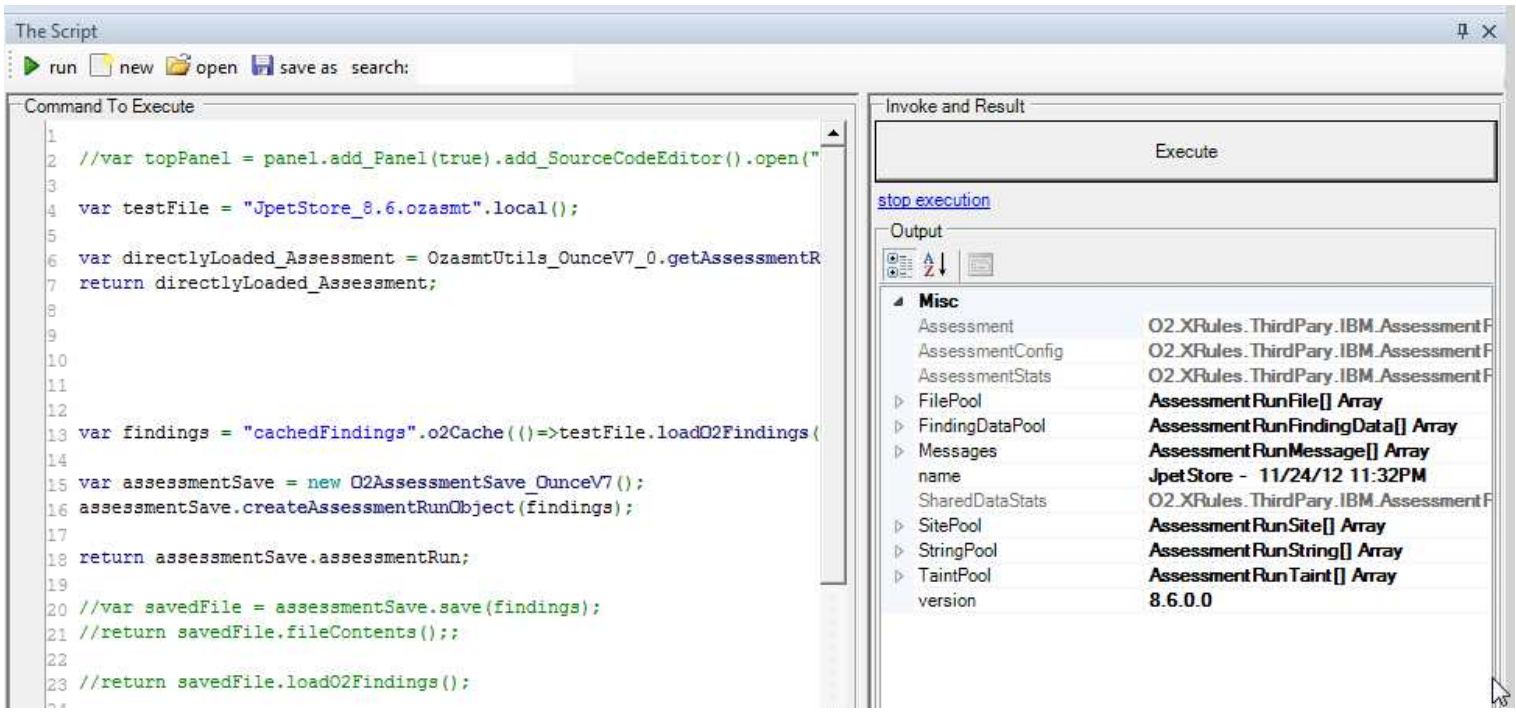
```
//var topPanel = panel.add_Panel(true).add_SourceCodeEditor().open("O2AssessmentSave_OunceV7_0.cs".local
());

var testFile = "JpetStore_8.6.ozasmt".local();
var findings = "cachedFindings".o2Cache(()=>testFile.loadO2Findings());

var assessmentSave = new O2AssessmentSave_OunceV7();
assessmentSave.createAssessmentRunObject(findings);

return assessmentSave.assessmentRun;

//var savedFile = assessmentSave.save(findings);
//return savedFile.fileContents();;

//return savedFile.loadO2Findings();


//"findings".popupWindow().add_FindingsViewer().show(findings);
//using O2.XRules.ThirdPary.IBM
//using O2.XRules.Database.Findings
//O2File:O2AssessmentSave_OunceV7_0.cs
//O2File:Findings_ExtensionMethods.cs
//O2File:xsd_Ozasmt_OunceV7_0.cs
```

The return object of the script above is the 7.x AssessmentRun (which exposes its structure)



here is how to load the ozasmt file directly (i.e. using the C#'s XSD)

```
var testFile = "JpetStore_8.6.ozasmt".local();

var directlyLoaded_Assessment = OzasmtUtils_OunceV7_0.getAssessmentRunObjectFromXmlFile(testFile);
return directlyLoaded_Assessment;
```

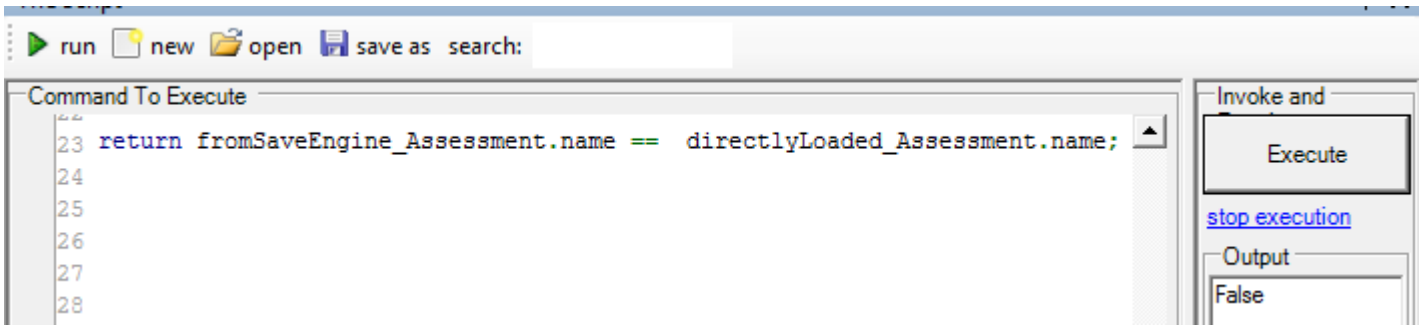


In order to make sure we are not losing (a lot) of important data, here is a way to compare them side-by-side

```
var testFile = "JpetStore_8.6.ozasmt".local();

Func<AssessmentRun> getVersionFromDirectLoad =
	()=>{
			return OzasmtUtils_OunceV7_0.getAssessmentRunObjectFromXmlFile(testFile);
		};

Func<AssessmentRun> getVersionFromSaveEngine =
	()=>{
			var findings = "cachedFindings".o2Cache(()=>testFile.loadO2Findings());
			var assessmentSave = new O2AssessmentSave_OunceV7();
			assessmentSave.createAssessmentRunObject(findings);
			return assessmentSave.assessmentRun;
		};

var directlyLoaded_Assessment = getVersionFromDirectLoad();

var fromSaveEngine_Assessment = getVersionFromSaveEngine();

return fromSaveEngine_Assessment;
```

for example the name at the moment doesn't match:

```
return fromSaveEngine_Assessment.name ==  directlyLoaded_Assessment.name;
```



;:

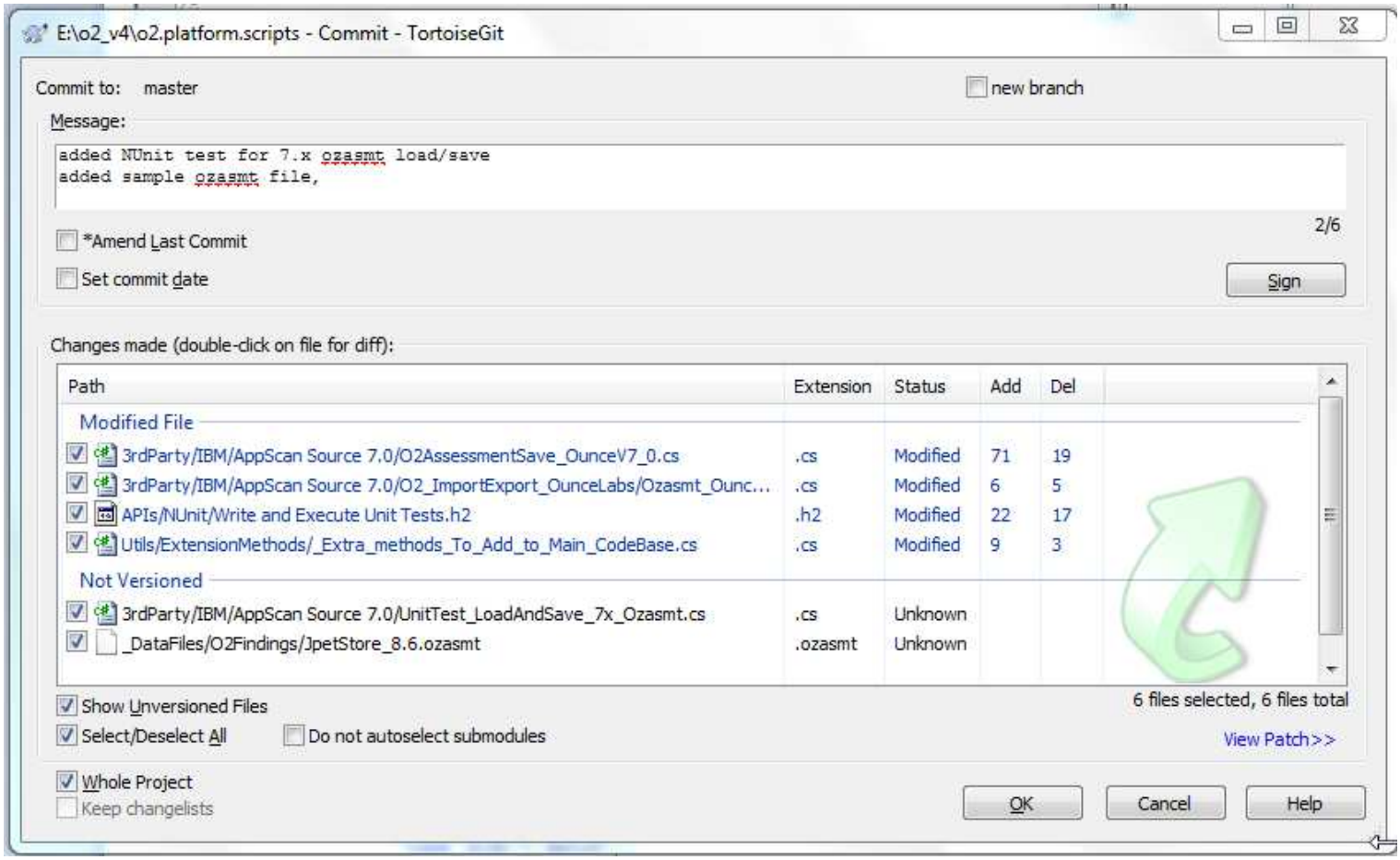
Next lets add the lambda methods to the Utils file (which is a test that at the moment fails)

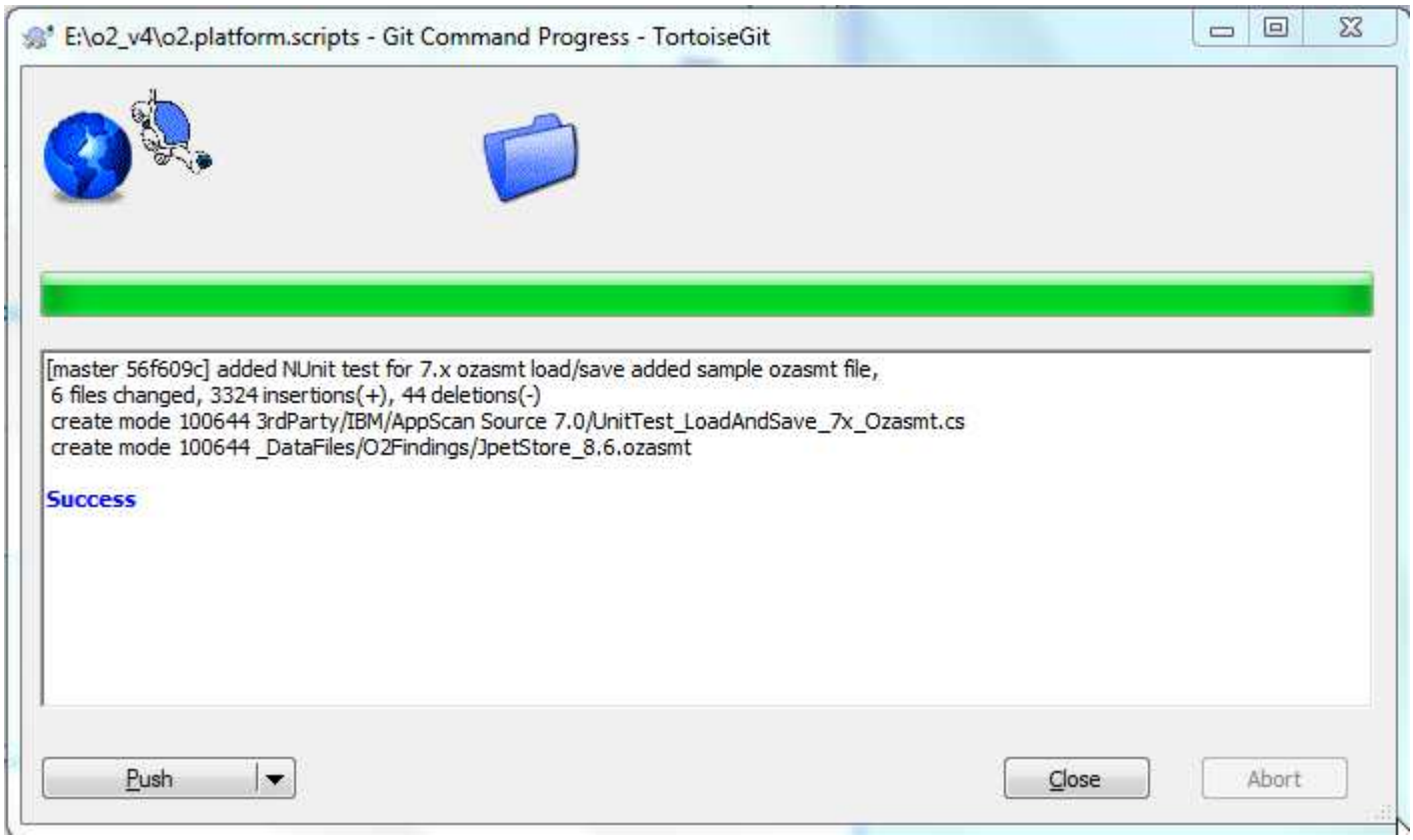
```
195
196     public class O2Assessment_OunceV7_Utils
197     {
198         public static AssessmentRun getVersionFromDirectLoad(string ozasmtFile)
199         {
200             return OzasmtUtils_OunceV7_0.getAssessmentRunObjectFromXmlFile(ozasmtFile);
201         }
202
203         public static AssessmentRun getVersionFromSaveEngine(string ozasmtFile)
204         {
205             var findings = ozasmtFile.loadO2Findings();
206             var assessmentSave = new O2AssessmentSave_OunceV7();
207             assessmentSave.createAssessmentRunObject(findings);
208             return assessmentSave.assessmentRun;
209         }
```

which will allow us to write a Unit test to check it:

```
[Test]
public void CheckAssessmentRunData_AfterLoad_and_InMemorySave()
{
    var directlyLoaded_Assessment = O2Assessment_OunceV7_Utils.getVersionFromDirectLoad(TestFile);
    var fromSaveEngine_Assessment = O2Assessment_OunceV7_Utils.getVersionFromSaveEngine(TestFile

    Assert.That(directlyLoaded_Assessment.name == fromSaveEngine_Assessment.name ,
                "name didn't match");
}
```

for reference here is the commit with the changes so far:




E:\o2_v4\o2.platform.scripts - Git Command Progress - TortoiseGit
[master 56f609c] added NUnit test for 7.x ozasmt load/save added sample ozasmt file,
6 files changed, 3324 insertions(+), 44 deletions(-)
create mode 100644 3rdParty/IBM/AppScan Source 7.0/UnitTest_LoadAndSave_7x_Ozasmt.cs
create mode 100644 _DataFiles/O2Findings/JpetStore_8.6.ozasmt
Success
Push
Close
Abort

.

..

.

.

.

.

.

.