

Israel Dot NET Developers User Group (IDNDUG)

November 2014 Meeting

>> Under the (hood of a) project

- * Multi- project-types Application
- * Platform Invocation Services (Pinvoke)

מספר מילים על המצגת!

מצגת זו הוכנה על ידי **רון אריאלי** לשימוש אישי!

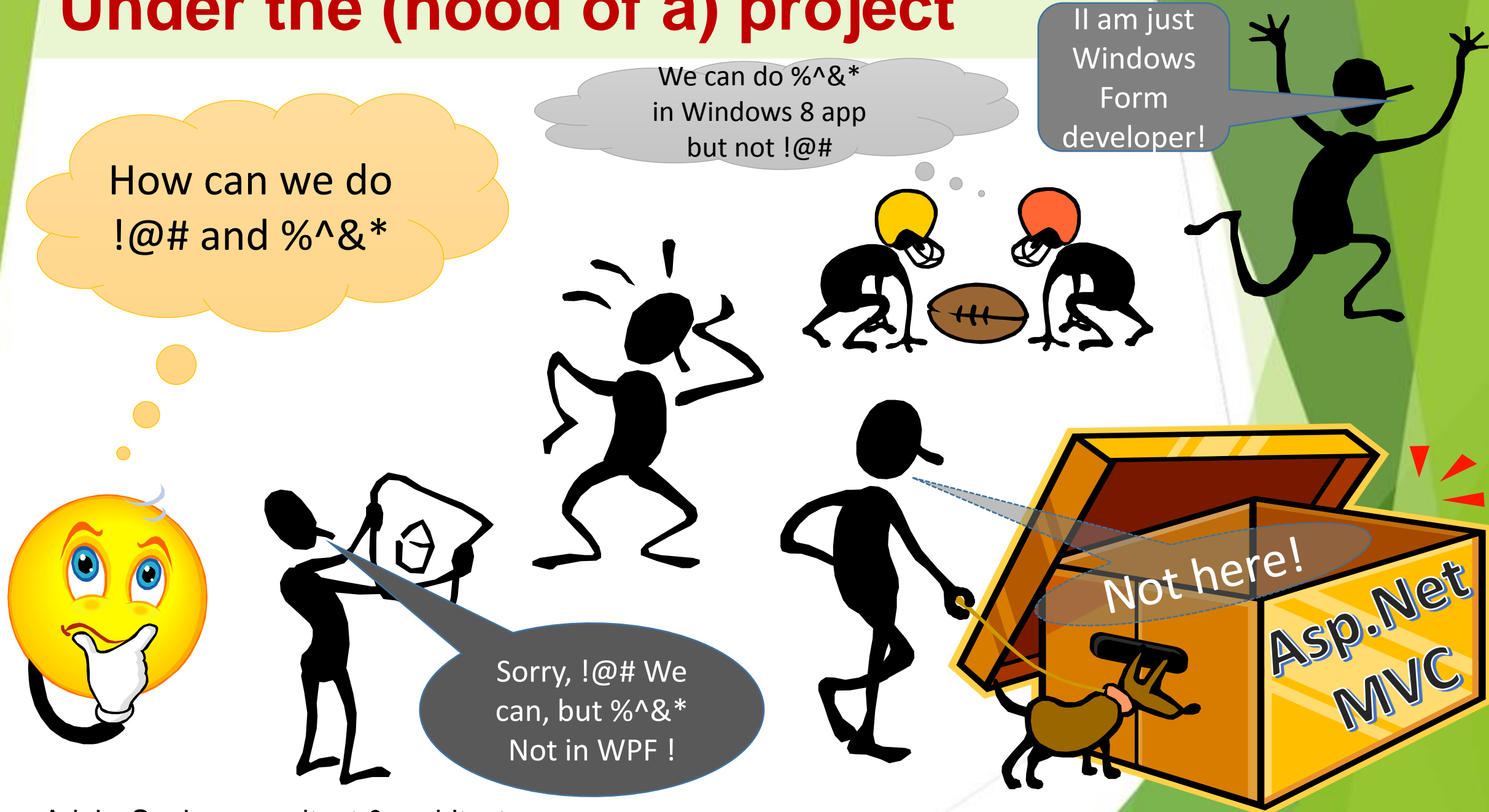
**כל שימוש במצגת או בתוכן המצגת, יעשה על אחריות המשתמש
בלבד!**

האישור להוריד מצגת זו נועד לשימוש אישי לבד, ואינו כולל אישור להפיץ את המצגת
בשום צורה שהיא!

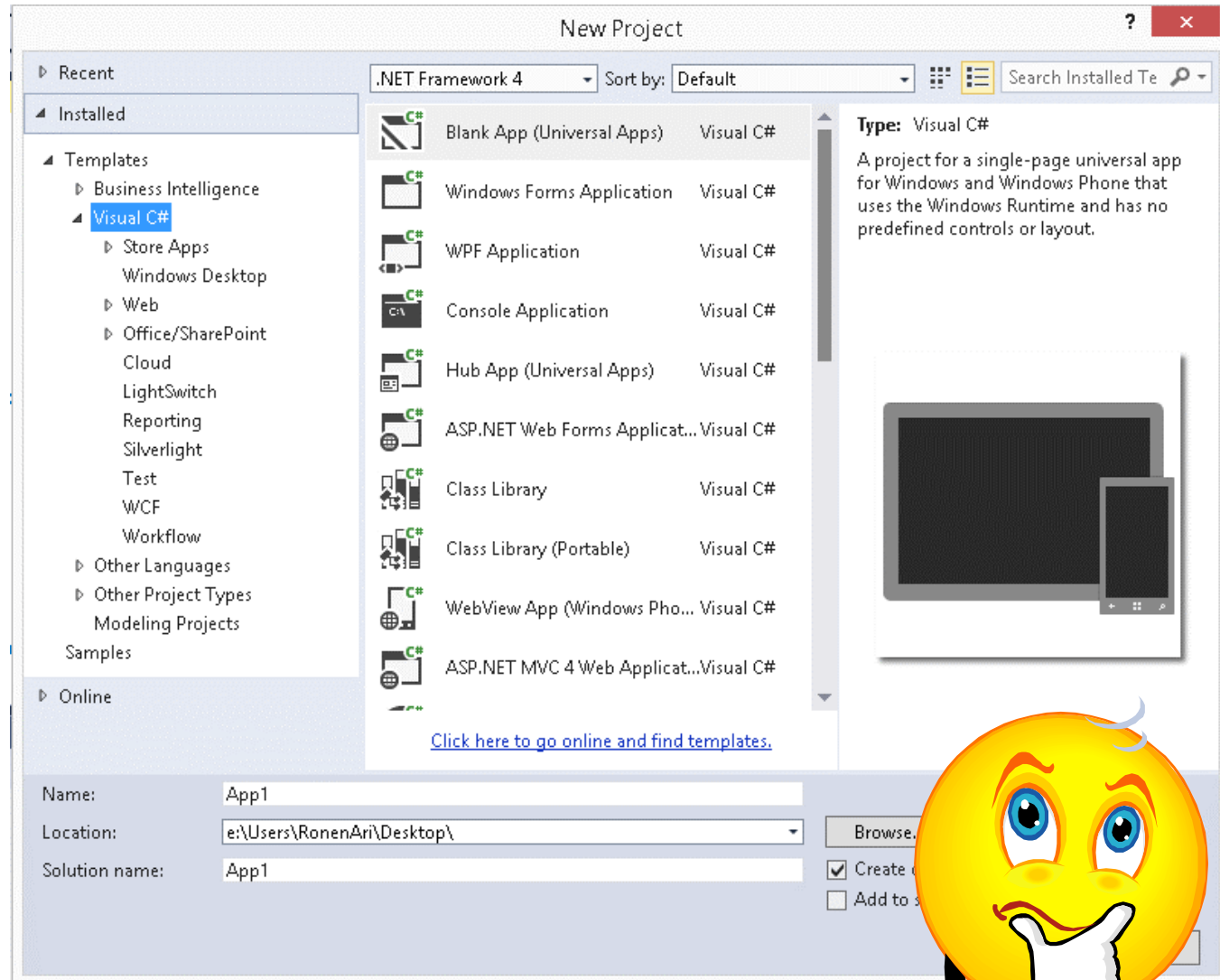
אם יש לכם פידבקים, הערות, או הארות, אז ניתן ליצור עמי קשר

דרך פרטי ההתקשרות באתר: <http://ariely.info>

Under the (hood of a) project



Under the (hood of a) project



There are hundreds
of projects in the
list...
**What do I need to
choose?!**



Multi-Type application

Ronen Ariely,
Senior consultant & architect

About

Ronen Ariely ([Blog](#), [website](#), [Facebook](#)) is a senior consultant & architect, a professional programmer, architecting SQL & BI enterprise level solutions. He has more than 15 years of experience in variety of programming languages and technologies, leading and managing development teams and BI projects. Ronen graduate his BA in Physics and continued MA studies by specialization in Geo-physics and seismic data analysis, at the Tel-Aviv University.

Ronen is also active in forums and communities in the field of Programming, SQL Server, T-SQL, under the nickname “pitoach” or “pituach”. Ronen is recognized by Microsoft due to his expertise and his community activities, and get to serve several years as Moderator at MSDN Israel communities. He is also got to represent us, and at this time, is probably the only Israeli, which serve as Moderator, at the MSDN Global communities. He is writing technical blogs and Microsoft TechNet articles, and he is one of the leaders of the TechNetWIKI global communities.



Key -> pitoach / pituach

Ronen Ariely,
Senior consultant & architect

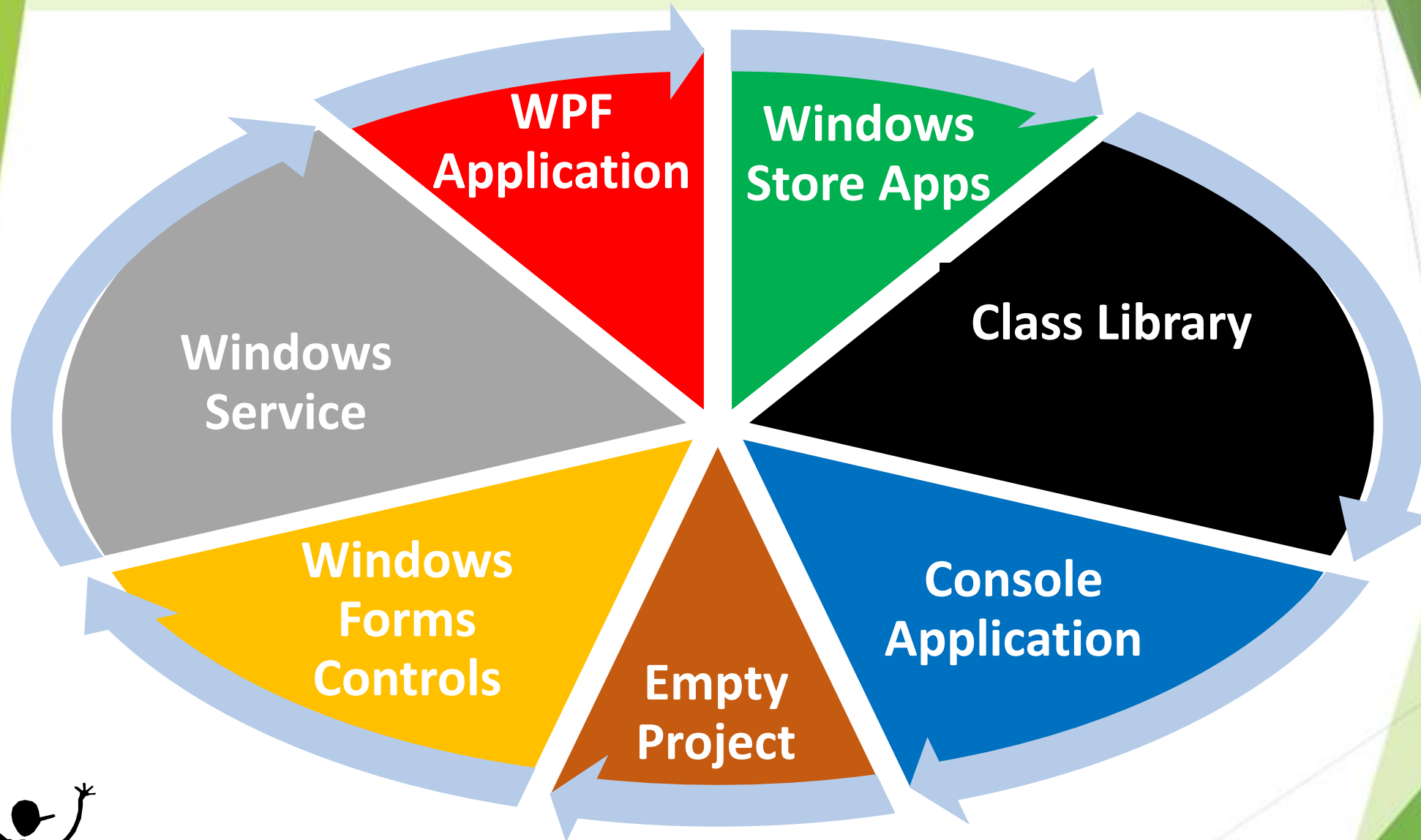
About

Ariel Ben Horesh is a well-known .NET expert, consultant, team leader and community leader (admin of the Israeli .NET User Group). With more than 15 years of experience in the software industry,

Ariel now works in CodeValue, a company he has co-founded, where he creates products for developers, consults and conducts courses around the world about UI development: WPF, Web, Mobile and UI architecture. Ariel is CodeValue's Mobile and Apps department manager, he has created well known apps in Windows Phone, Windows 8.1, iOS and other mobile platforms.

Ariel serve as a co-leader for the Israel Dot NET Developers User Group (IDNDUG), along with Shay Bar-Lev.

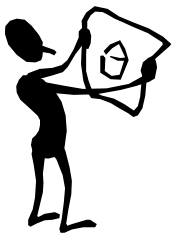
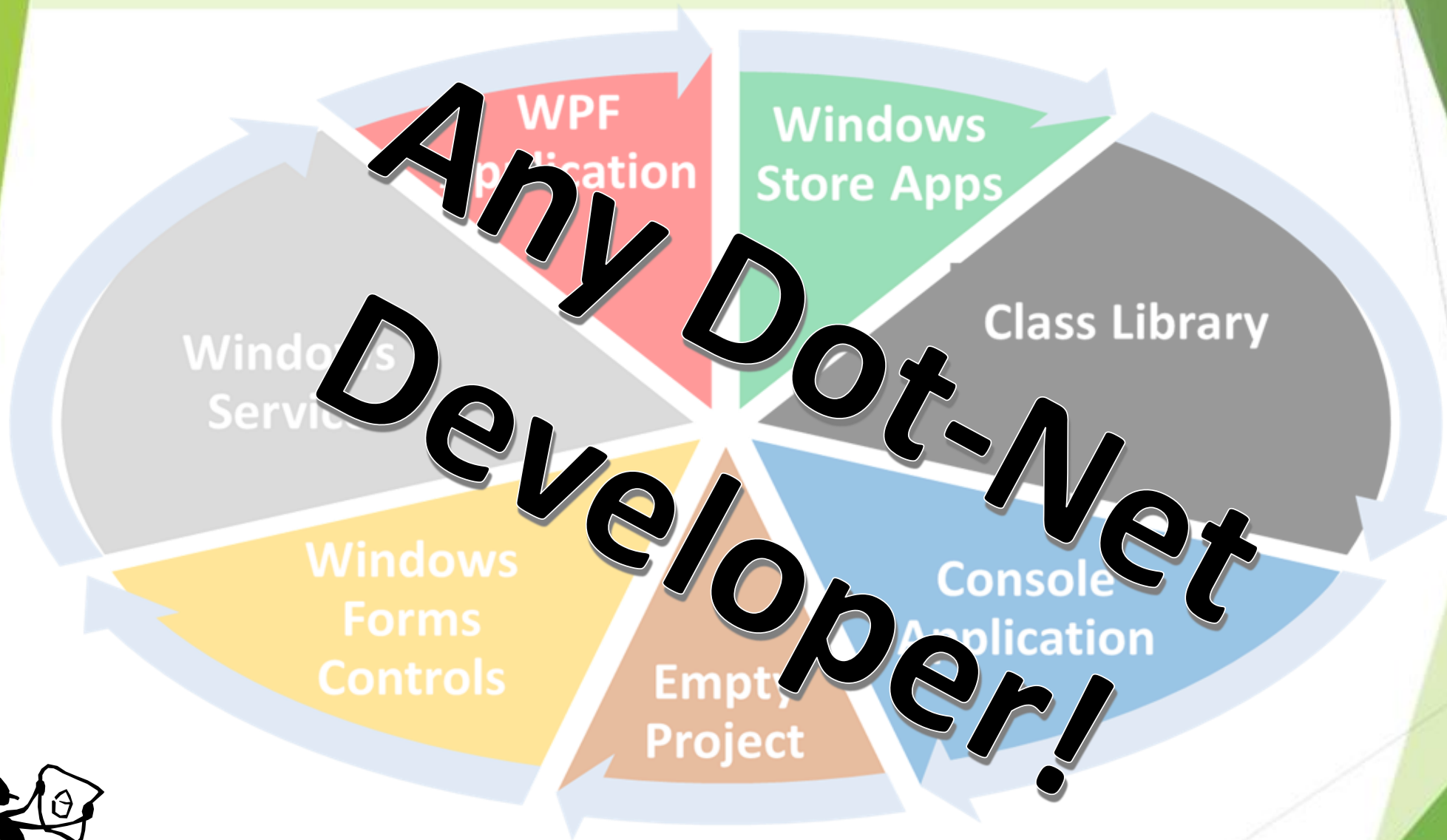
Target Audience



New

Ronen Ariely,
Senior consultant & architect

Target Audience



New

Ronen Ariely,
Senior consultant & architect

Agenda

>> Under the (hood of a) project

The project...

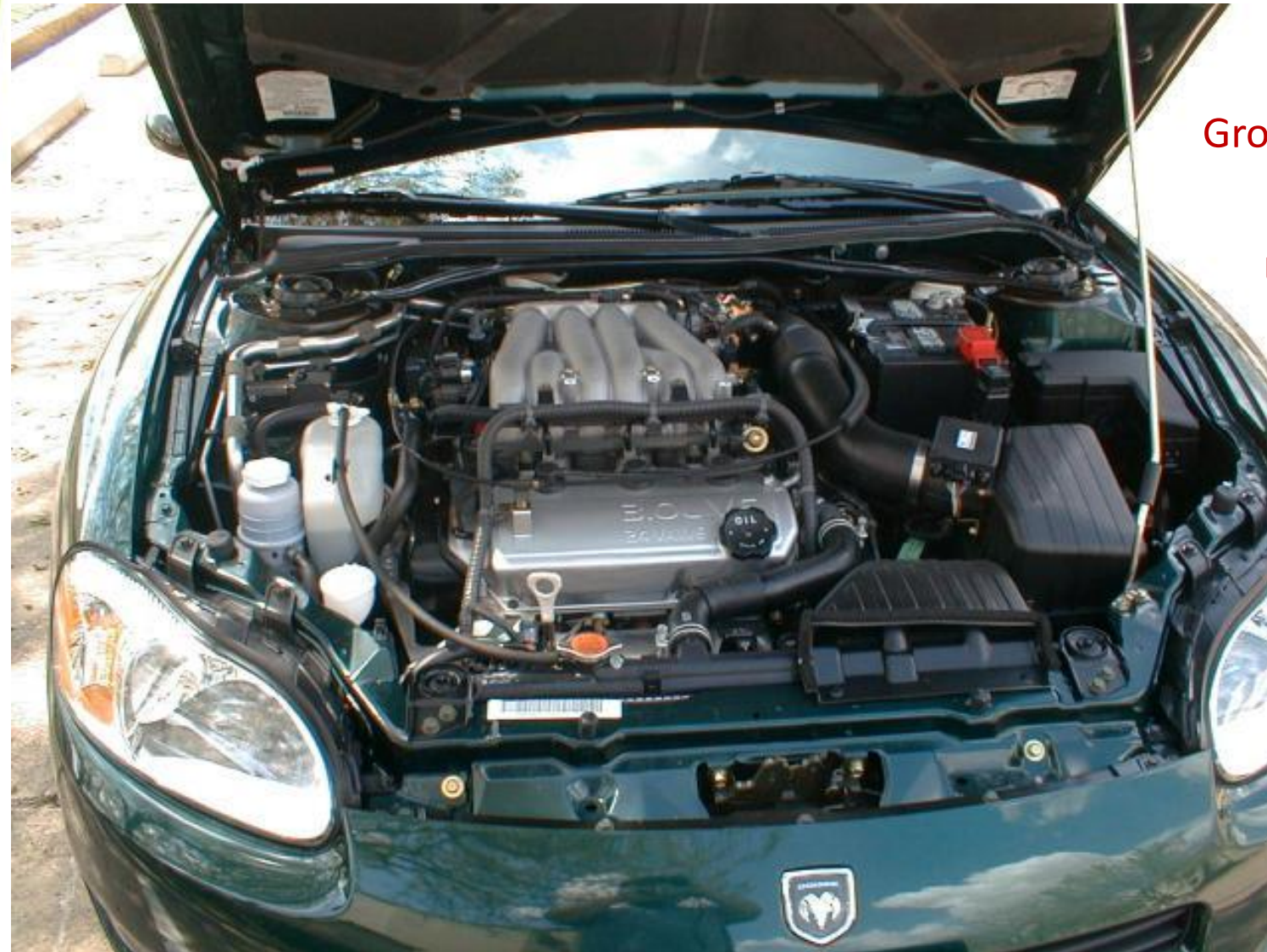
Before, During, and After...

What is behind

>> Multi-types Application

>> Platform Invocation Services (Pinvoke)

Under the (hood of a) project



Ground Level

Under the hood

Behind the scene

Three levels, From the GUI to the IL throw The Templates

Under

The

*(hood
of a)*

project

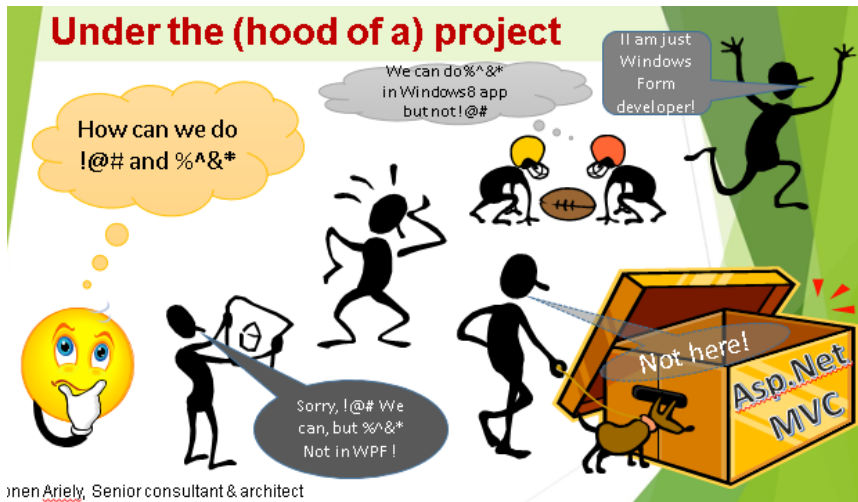
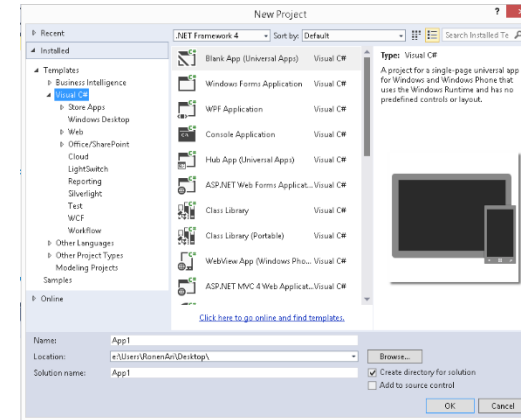
Ronen Ariely,
Senior consultant & architect

Under the (hood of a) project

Creating a new project, Using the Visual Studio

>> Microsoft:

The project **template** that you choose determines the output type and other options available for that project.



ronen Ariely, Senior consultant & architect

=> Question:

So what if we need to develop a project, which use options, from several build-in templates?

Templet, for Executable App

Under

The

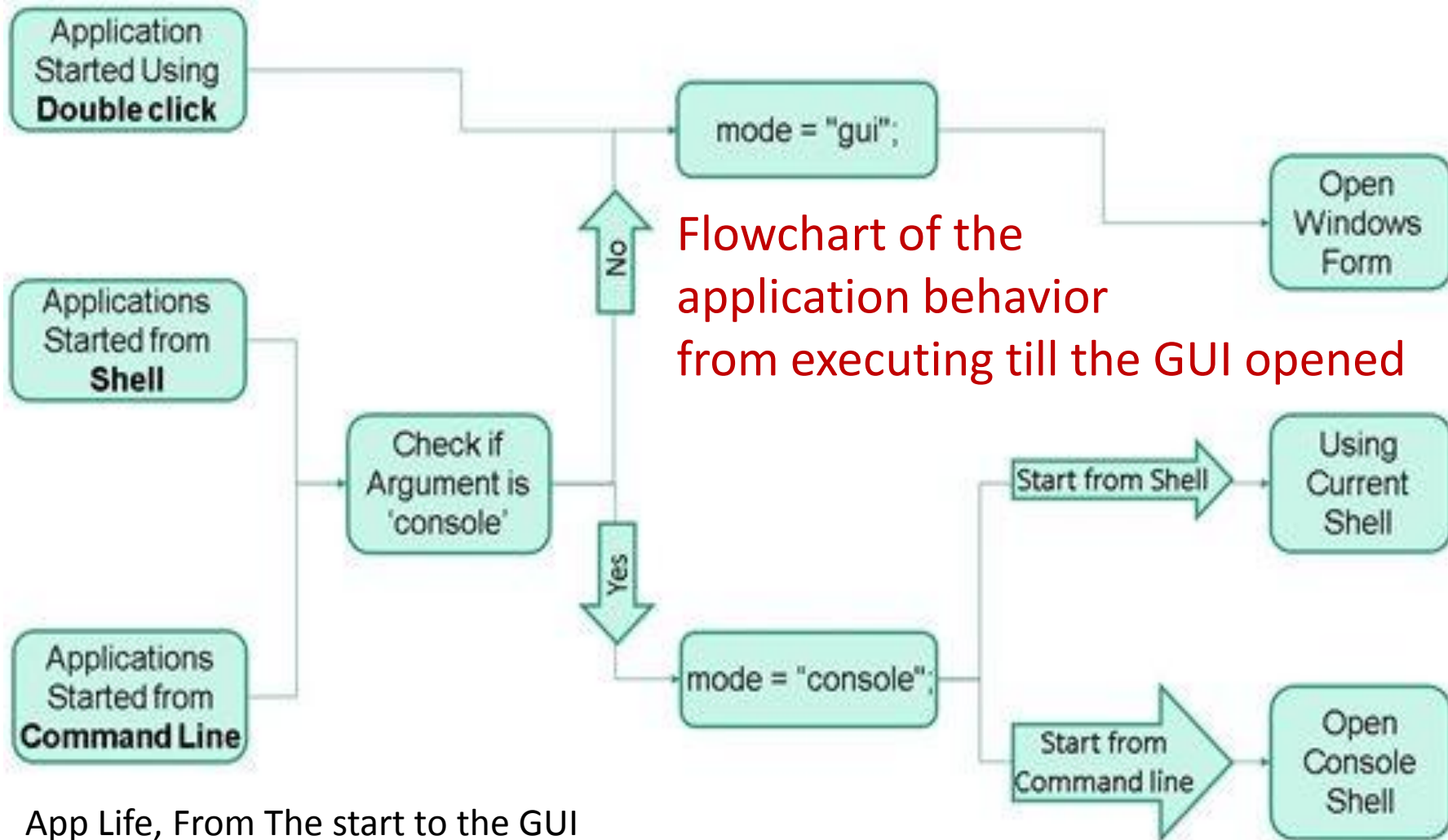
(hood of a)

project

Ronen Ariely,
Senior consultant & architect

Under the (hood of a) project

Multi-Types Application



App Life, From The start to the GUI

Under

The

(hood of a)

project

Ronen Ariely,
Senior consultant & architect

Under the (hood of a) project

Entry Point



*Under
The
(hood
of a)
project*

Start ?!?! ---> Entry Points or Point? C#-> Main, but .Net?!?! ---> IL .entrypoint

Ronen Ariely,
Senior consultant & architect

Under the (hood of a) project

Entry Point

>> Entry point is where control enters a program or piece of code. It is where a program starts its execution.

In most of today's popular computer systems (Windows, Unix), a computer program usually only has a single entry point. Android applications might have multiples entry points.

In C, C++, D and programs this is a function named main; in Java and C# (.Net languages) it is a static method named **main**

The CLR searches for a static method marked with the .entrypoint IL directive, which takes either no arguments, or a single argument of type string[], and has a return type of void or int, and executes it

*Under
The
(hood
of a)
project*

Under the (hood of a) project

Entry Point

>> The code that we write in a .Net programming language like C#, is finally converted to either Assembler or Intermediate Language (IL).

>> Microsoft has provided a program called ilasm whose sole task is to create an executable file from an IL file.

```
a.ll
.assembly mukhi {}
.method void vijay()
{
    .entrypoint
    ret
}
```

Under

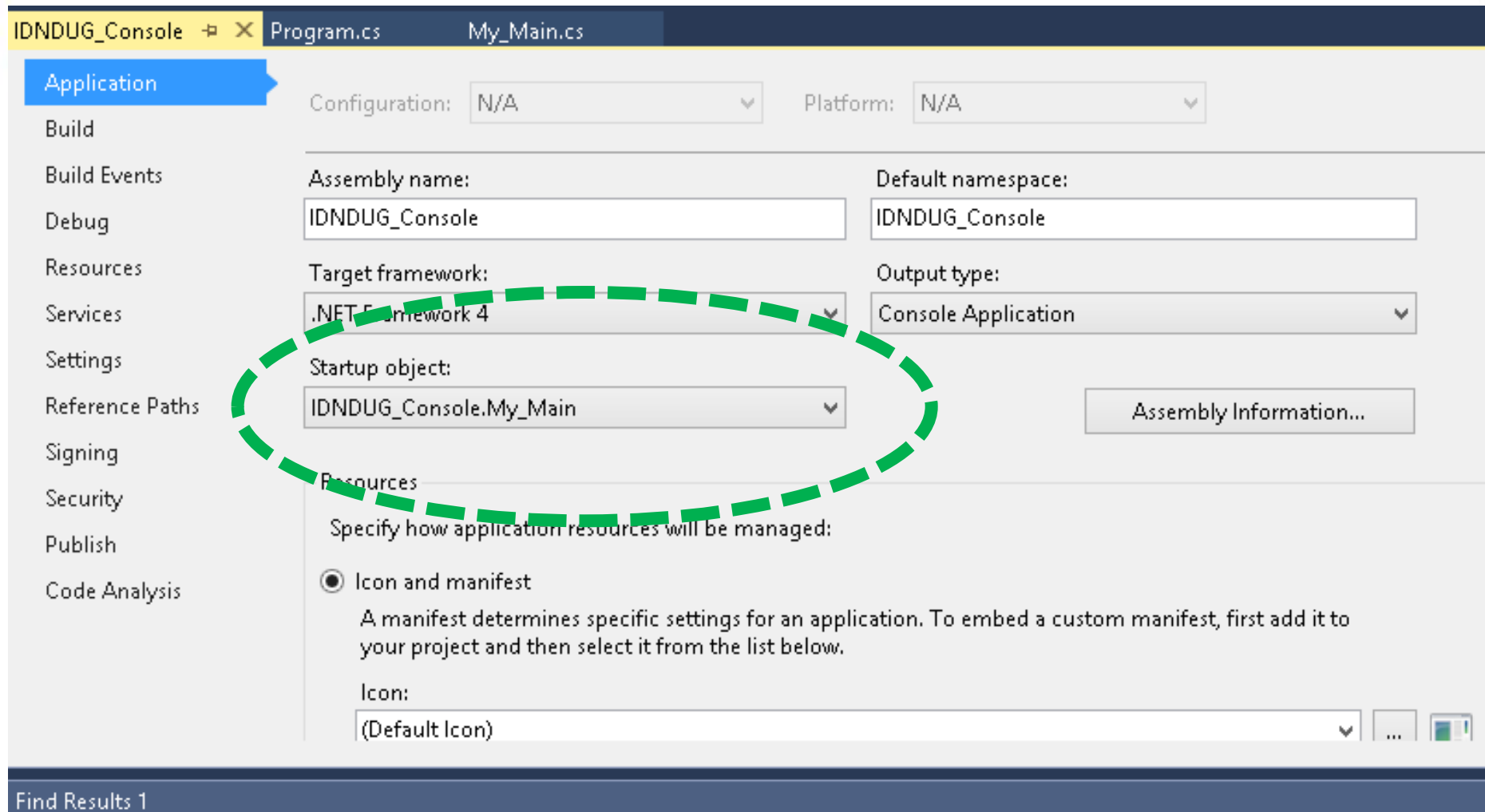
The

*(hood
of a)*

project

Under the (hood of a) project

Entry Point



*Under
The
(hood
of a)
project*

C Sharp: Multi-Types Application

Play Time

Executing one application in variety ways, on variety interfaces. For example: (1) command line (with or without parameters), or by (2) double click the exe file, or (3) from command shell / PowerShell. Moreover, using variety of Graphical user interfaces...

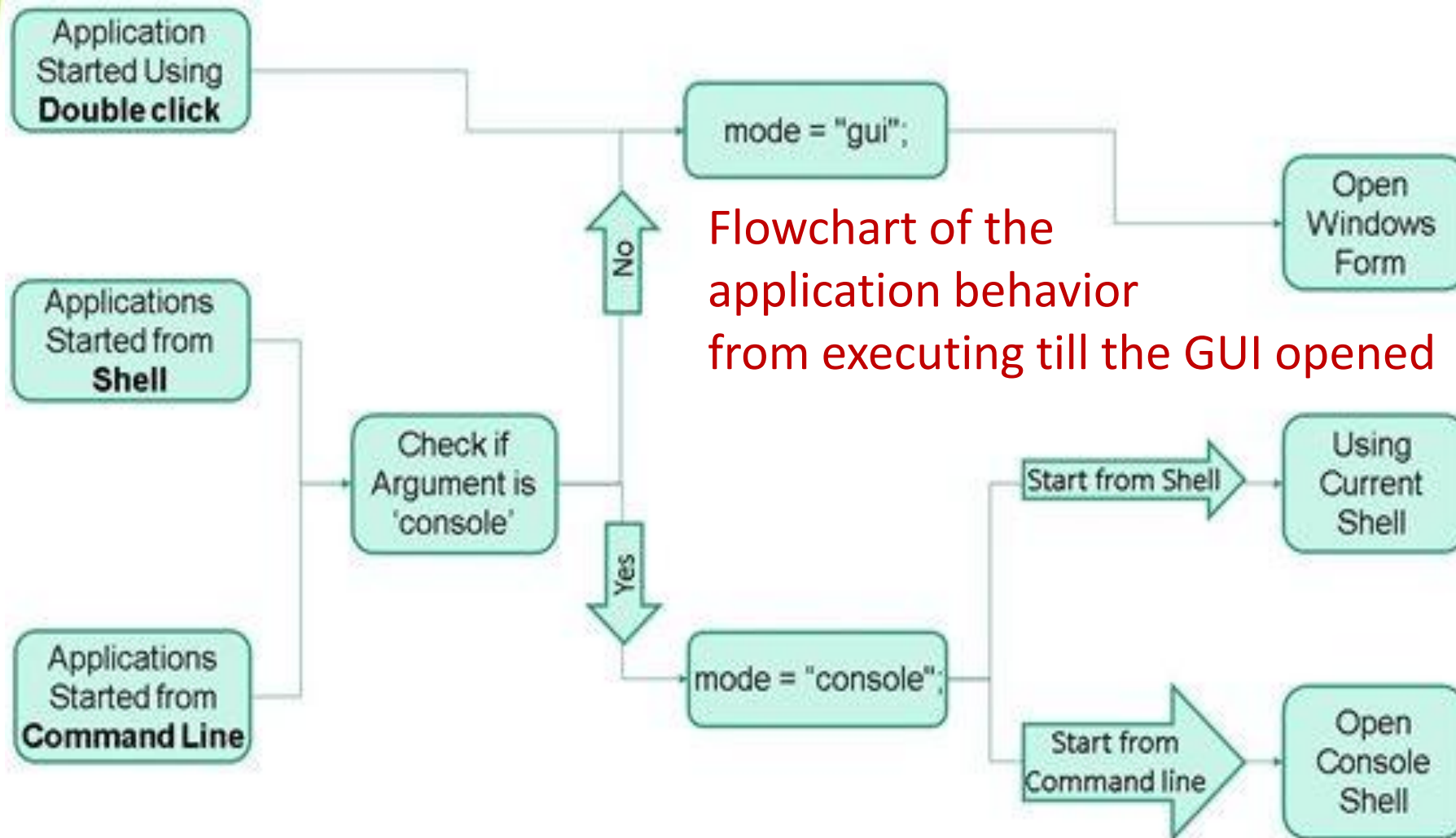
>> <http://social.technet.microsoft.com/wiki/contents/articles/25700.c-sharp-multi-types-application.aspx>

>> <https://gallery.technet.microsoft.com/C-Sharp-Multi-Types-dceecbce>

*Under
The
(hood
of a)
project*

Under the (hood of a) project

Multi-Types Application



Under

The

(hood of a)

project

חידה

ניתן לחלק את כל האפליקציות של דוט-נט ל 2 קבוצות מרכזיות בהקשר של נקודת ההתחלה.

* אני לא מדבר על תבניות שונות אלא התנהגויות שונות, בהקשר של נקודת הכניסה.

רמז: בדמו הקודם דיברנו על תבניות הקשורות לאפליקציות מהקבוצה הראשונה. מה הקבוצה הראשונה ומה השנייה?

יש רעיון מה הן הקבוצות?

*Under
The
(hood
of a)
project*

Under the (hood of a) project

Dynamic Entry point method

Extensible Application Markup Language

XMAL elements map directly to Common Language Runtime object instances, while XAML attributes map to Common Language Runtime properties and events on those objects..

Under

The

*(hood
of a)*

project

Communicating between classes using MVVM Light Messenger

Play Time

This sample is intended for developers working in Winforms, WPF, Silverlight, Windows phone or Windows store app. The solution demonstrates an approach to communicating between disconnected pieces of code using MVVM Light Messenger.

>> <http://social.technet.microsoft.com/wiki/contents/articles/26070.communicating-between-classes.aspx>

>> <https://gallery.technet.microsoft.com/Communicating-between-eadd027c>

* This application bring us an example of using Windows form project with WPF

*Under
The
(hood
of a)
project*

חידה

מה ההבל בין קובץ MSI לקובץ EXE שמבצע התקנה (פרויקט שיוצר קבצים
אלו עבור התקנה)

*Under
The
(hood
of a)
project*

Under the (hood of a) project

Installing with MSI vs Exe

EXE file is an application, contain executable code for a computer to run, and can do anything.

MSI files used by Windows Installer application. It is a Setup Package. It contain features and components information about an application. It may hold files, registry data, shortcuts etc. It contain Information for the installation: UI, prerequisites, actions to execute, order procedure, whether to support Administrative installations, etc.

>> <http://channel9.msdn.com/Forums/Coffeehouse/30103-EXE-vs-MSI/88792c9ac2cc4dadbbc79dea01368b03>

Under

The

*(hood
of a)*

project

Under the (hood of a) project

Pinvoke - Platform Invoke

Platform Invocation Services allows managed code to call unmanaged functions that are implemented in a DLL.

Directly call unmanaged code:

- * Directly call a function exported from a DLL.
- * Call an interface method on a COM object.

```
using System.Runtime.InteropServices;
```

```
[DllImport("msvcrt.dll")]  
public static extern int puts(string c);  
[DllImport("msvcrt.dll")]  
internal static extern int _flushall();  
  
public static void Main() {  
    puts("Test"); _flushall();  
}
```

Inside the
CLASS

Under

The

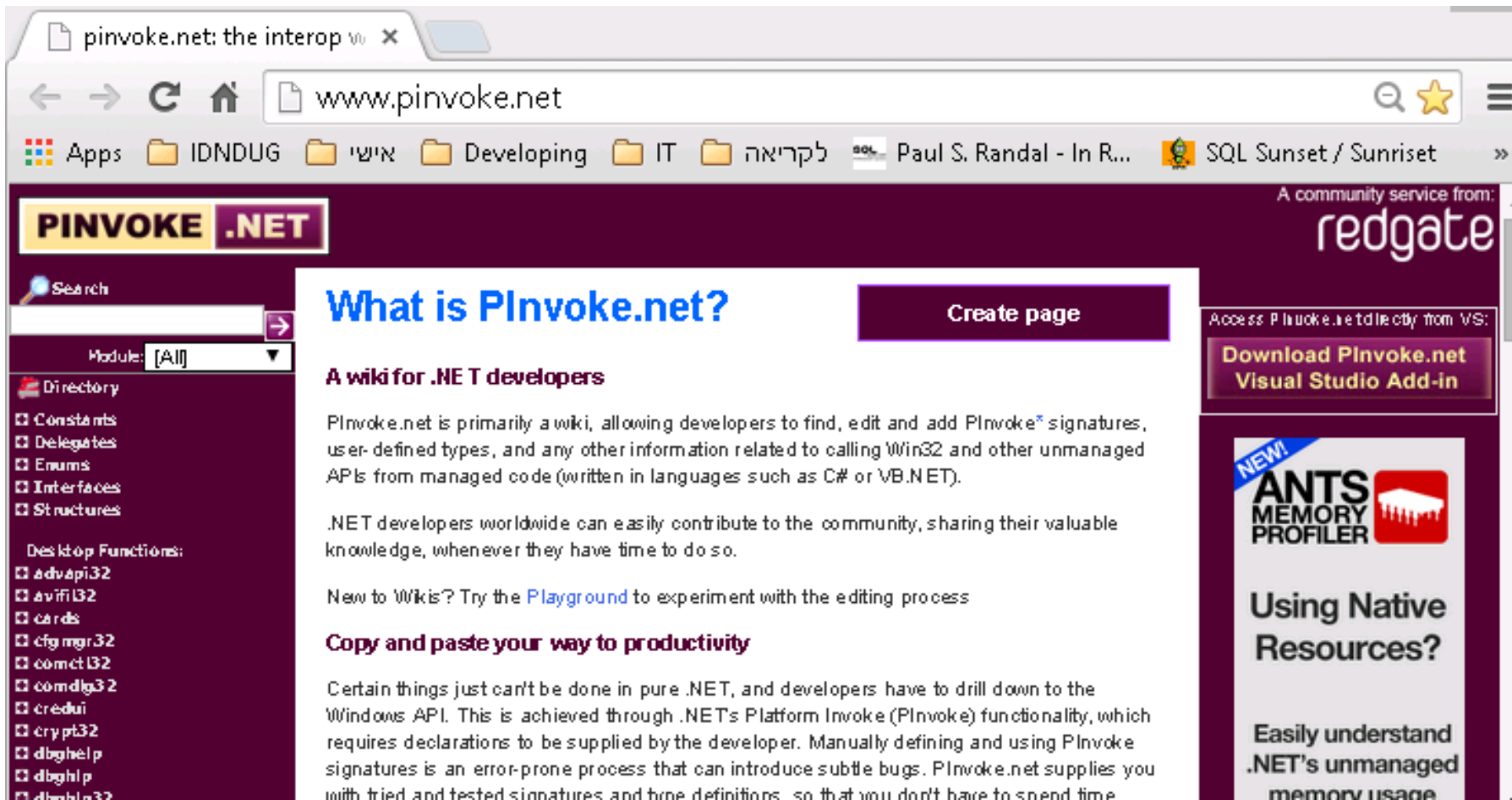
(hood
of a)

project

Under the (hood of a) project

Pinvoke - Platform Invoke

<http://www.pinvoke.net/>



Under

The

(hood of a)

project

Ronen Ariely,
Senior consultant & architect

Under the (hood of a) project

Play Time

Communicating between applications using Pinvoke

>> <https://github.com/arielbh/PInvoke-Samples>

More Pinvoke examples

*Under
The
(hood
of a)
project*

Under the (hood of a) project

Pinvoke

Under

The

(hood
of a)

project

עתה אפשר לחזור ולעבור על השימוש שעשינו ב Pinvoke בדמו הראשון

* GetForegroundWindow: על מנת לקבל מצביע אל החלון עליו עובדים.

```
IntPtr ptr = GetForegroundWindow();
```

* GetWindowThreadProcessId: קבלת ה ID התהליך שלנו בעזרת המצביע.

```
GetWindowThreadProcessId(ptr, out u);
```

* AttachConsole: פתיחת CONSOLE ב SHEL הנוכחי.

```
AttachConsole(process.Id);
```

* AllocConsole: פתיחת GUI חדש של CONSOLE.

```
AllocConsole();
```

* FreeConsole: בסיום נשחרר את ה SHELL שלנו

```
FreeConsole();
```

Under the (hood of a) project

Conclusions and summary

<< התחלנו בהצגת אמירה "בלתי אפשרי"

<< הדגשנו שאנחנו בסך הכול עושים שימוש בתבניות מוכנות בפתיחת פרויקט.
-- צורך למיטוב תהליך הפיתוח

<< ראינו שהתבנית לא חייבת להיות מגבלה, וניתן להרחיב או לשנות אותה.
-- יכולות להיות מגבלות הקשורות לסביבת ההרצה (למשל מערכת ההפעלה).
-- אל תמהרו לענות או לקבל תשובה "בלתי אפשרי" בגלל סוג האפליקציה.

<< הצגנו שילוב אלמנטים מתבניות שונות של .Net. תחת פרויקט בודד (לא SOLTION).

<< עברנו על שילוב פונקציות מקוד לא מנוהל Pinvoke

<< בהרצאה התעמקנו בשלבי החיים הראשונים של אפליקציה מההרצה ועד הפעלת ה GUI.
-- עברנו על נושא Entry Point

-- התעמקנו רק בפרויקטים שנועדו ליצירת יישום הרצה, ולא בספריות קוד.

Under

The

(hood
of a)

project

Under the (hood of a) project

Conclusions and summary

<< התחלנו בהצגת אמירה "בלתי אפשרי"

<< הדגשנו שישנו בסך הכל שימושי בתבניות מוכנות בפתיחת פרויקט.
-- צורך למי שחושבים שהם יודעים הכל

<< ראינו שהתבנית לא חייבת להיות מגבלה, וניתן להרחיב או לשנות אותה.
-- יכולות להיות מגבלות הקשורות לכביש ההרצה (למשל מערכת ההפעלה).
-- אל תמהרו לענות או לקבל תשובה בלתי נכונה "בגלל סוג האפליקציה".

<< הצגנו שילוב אלמנטים מתבניות שונות של .Net תחת פיקוד (למשל SOLTIO).

<< עברנו על שילוב פונקציות מקוד לא מנוהל Pinvoke

<< בהרצאה התעמקנו בשלבי החיים הראשונים של אפליקציה מההרצה ועד הפעלת ה GUI.

-- עברנו על נושא Entry Point

-- התעמקנו רק בפרויקטים שנועדו ליצירת יישום הרצה, ולא בספריות קוד.

Under

The

(hood
of a)

project

Under the (hood of a) project

Sources and Resources

*Under
The
(hood
of a)
project*

Under the (hood of a) project

Sources and Resources

- Project Properties (Visual Studio)
[http://msdn.microsoft.com/en-us/library/vstudio/c7f8bfy4\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/c7f8bfy4(v=vs.100).aspx)
- Default Project Templates in Visual Studio
[http://msdn.microsoft.com/en-us/library/vstudio/0fyc0azh\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/0fyc0azh(v=vs.100).aspx)
- Specifying an Entry Point
[http://msdn.microsoft.com/en-us/library/f5xe74x8\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/f5xe74x8(v=vs.110).aspx)
- Main() and Command-Line Arguments (C# Programming Guide)
<http://msdn.microsoft.com/en-us/library/acy3edy3.aspx>
- Extensible Application Markup Language
http://en.wikipedia.org/wiki/Extensible_Application_Markup_Language

Under

The

*(hood
of a)*

project

Under the (hood of a) project

Sources and Resources

- C Sharp: Multi-Types Application
<http://social.technet.microsoft.com/wiki/contents/articles/25700.c-sharp-multi-types-application.aspx>
- Closing Main Form Without exit the application
<http://ariely.info/Blog/tabid/83/EntryId/126/Closing-Main-Form-Without-exit-the-application.aspx>
- "Transparent" application (application without GUI)
<https://social.msdn.microsoft.com/Forums/he-IL/9eb436db-69a7-44f0-81aa-bc79b15fd69d/>
- The MVVM Light Messenger In-Depth
<http://msdn.microsoft.com/en-us/magazine/dn745866.aspx>
- Introduction to Microsoft's IL
<http://www.vijaymukhi.com/documents/books/ilbook/chap1.htm>

Under

The

*(hood
of a)*

project

Under the (hood of a) project

Sources and Resources

- Platform Invoke Tutorial
[http://msdn.microsoft.com/en-us/library/aa288468\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa288468(v=vs.71).aspx)
- Calling Win32 DLLs in C# with P/Invoke
<http://msdn.microsoft.com/en-us/magazine/cc164123.aspx>
- Working with Win32 API in .NET
<http://www.c-sharpcorner.com/UploadFile/shrijeetnair/win32api12062005005528AM/win32api.aspx>
- <http://www.pinvoke.net/>
- <http://www.codeproject.com/Articles/403285/P-Invoke-Tutorial-Basics-Part>
- [http://msdn.microsoft.com/en-us/library/aa288468\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa288468(v=vs.71).aspx)
- Communicating between applications using Pinvoke
<https://github.com/arielbh/PInvoke-Samples>

*Under
The
(hood
of a)
project*