

Создание плагина для Robin Studio Выпуск

ROBIN RPA Team

дек. 24, 2020

Содержание:

 $\mathbf{2}$

1 Создание плагина для Robin Studio



Глава 1

Создание плагина для Robin Studio

Для того чтобы добавить новый плагин в проект вам нужно проделать следующие шаги:

1. Открыть проект плагинов нажать по Solution правой кнопкой мыши (ПКМ) \rightarrow Add \rightarrow New Project...:

			Sear	ch Solution Explorer (Ctrl+;)	- م
	Build Solution Rebuild Solution Clean Solution Analyze Batch Build Configuration Manager Manage NuGet Packages for Solution Restore NuGet Packages New Solution Explorer View Calculate Code Metrics Project Dependencies Project Build Order	Ctrl+Shift+E	•	Solution 'plugins' (21 projects) 1 C# ChatBotKafka DataConversion E Dialogs E Email E Excel FilesSystem FTP-SFTP GoogleTables Outlook PDF Preferentum Preferentum Queue Queue Security	Ì
2	Add			C# SimulateSystem	
‡ ©	Set StartUp Projects View History			Existing Project Existing Web Site	
ی م	Compare with Unmodified Blame (Annotate)			* New Item Ctrl+Shift+A T Existing Item Shift+Alt+A New Solution Folder	• # × •
x X	Rename	Ctri+v		isc	
0	Open Folder in File Explorer			ame) plugins	
ىر	Properties	Alt+Enter		tive config Debug Win32	

Рис.1 Добавление нового плагина в проект Plugin

2. Из списка предлагаемых шаблонов для создания проектов выбрать Class Library (.NET Framework). В строке Name написать название вашего плагина и выбрать Framework(если конечно

Add New Project					8 23
▶ Recent	4	Sort by:	Default 🔹 🎬 🗄		class library c# 🛛 🗙 👻
▲ Installed		œ	WCF Service Library	- Visual C#	Type: Visual C#
✓ Visual C# Get Started Windows Unive Windows Deskt ▷ Web	ersal top		Class Library (.NET Core) Class Library (.NET Standard)	Visual C# Visual C#	A project for creating a C# class library (.dll)
Office/SharePo NET Core	int		Class Library (.NET Framework) 1	Visual C#	
.NET Standard Android			Class Library (Legacy Portable)	Visual C#	
Apple TV Apple Watch			Class Library (Universal Windows)	Visual C#	
Cloud Cross-Platform			Class Library (For U-SQL Application)	Azure Data Lake	
Extensibility iOS Extensions		€ •3	iOS Class Library (Xamarin)	Visual C#	
iPhone & iPad Test			Android Class Library (Xamarin)	Visual C#	
WCF Workflow			iOS Bindings Library (Xamarin)	Visual C#	
♦ Visual Basic ♦ Visual C++			Android Bindings Library (Xamarin)	Visual C#	
Not finding what yo	u are looking for?				
Name	Tect				
Location: E:\Robin\robinRC\plu		olugins			Browse
Framework:	.NET Framework 4.	7.1 -			
					4 OK Cancel

он уже не стоит по умолчанию): .NET Framework 4.7.1. Далее жмем ОК:

Рис.2 Выбор шаблона «Class Library»

3. Кликаем ПКМ по только что созданному проекту (в данном случае Test)→ Properties:

Рис.3 Свойства плагина

4. Появится окно со свойствами вашего проекта:

Рис.4 Свойства плагина по дефолту

На вкладке Application нужно поменять следующие поля:

```
Assemblyname: Test Abbin.Test,
Default namespace: Test Robin.RobotTools,
И на всякий случай перепроверить Target fraemwork должен быть: .NET Framework 4.7.1.
```

В итоге должно получится:

Рис.5 Отредактированные свойства плагина

Coxpaните изменения (Ctrl+S).

5. Переименуйте класс "Class1" в "ActionLibrary" и 2 папки со следующим наименованием: Actions и Studio. Должно выглядеть следующим образом:

Рис.6 Дефолтный вид проекта плагина

6. Добавьте в проект референсы System.ComponentModel.DataAnnotations и из Manage NuGet Packages... Скачайте пакет Studio.Core, log4net, Robot.Core:

Рис.7 Добавление референсов в проект плагина

Если ссылки на пакеты добавились в packages.conf, перенести их в проджект-референс.

7. Правой кнопкой нажмите по Properties $\rightarrow \texttt{Add} \rightarrow \texttt{New Item...}$

⊿ ⊕	C# Test 1	
*	Build	
	Rebuild	
	Clean	
	Analyze	۱.
2	Publish Stored Procedures	
	Scope to This	
	New Solution Explorer View	
	Build Dependencies	Þ
	Add	۱.
Ĕ	Manage NuGet Packages	
₽	Set as StartUp Project	
	Debug	Þ
	Initialize Interactive with Project	
ጽ	Cut	Ctrl+X
பி	Paste	
×	Remove	Del
X	Rename	
	Unload Project	
ç	Open Folder in File Explorer	
للح	Properties 2	Alt+Enter

Build Events	Assembly name:	Default namespace:	
Debug	Test	Test	
esources	Target framework:	Output type:	
Services	.NET Framework 4.7.1	▼ Class Library	
Settings	Auto-generate binding redirects		
Reference Paths	Startup object:		
Signing	(Not set)	 Assembly Information 	
Code Analysis			
	Resources		
	Specify how application resources will be	managed:	
	 Icon and manifest 		
	A manifest determines specific setting your project and then select it from the	js for an application. To embed a custom manifest, first add it to e list below.	
	Icon:		
	(Default Icon)	▼ Browse	

Application*	Configuration: N/A	form: N/A
Build*		
Build Events	Assembly name:	Default namespace:
Debug	Robin.Test	Robin.RobotTools
Resources	Target framework:	Output type:
Services	.NET Framework 4.7.1	Class Library 🔹
Settings	Auto-generate binding redirects	=
Reference Paths	Startup object:	
Signing	(Not set) 🗸	Assembly Information
Security		
Publish	Resources	
Code Analysis	Specify how application resources will be managed:	
	Icon and manifest	
	A manifest determines specific settings for an app	olication. To embed a custom manifest, first add it to \mathbf{v}
	•	4



🔺 +C# Te	est
	Properties
⊮	References
	🗗 Analyzers
4	🍖 log4net
l	■ Microsoft.CSharp
3	🍖 Robot.Core
2	🍖 Studio.Core
l	■■ System
1	System.ComponentModel.DataAnnotations
I	■■ System.Core
	∎•■ System.Data
	System.Data.DataSetExtensions
	■■ System.Net.Http
	■■ System.Xml



Рис.8 Добавление в Properties нового ресурса

Находим шаблон Resources File и выбираем. В поле Name пишем следующие имя: Strings.resx и нажимаем Add:

Add New Item - Test				? 💌
 Installed 	Sort by:	Default 👻 🏭		re × •
✓ Visual C# Items WPF	⊕	ASP.NET Module	Visual C# Items	Type: Visual C# Items A file for storing resources
Code Data	Į,	Custom Control	Visual C# Items	
General SQL Server	Ŷ	editorconfig File (.NET)	Visual C# Items	
Storm Items ▷ Test	Ŷ	editorconfig File (default)	Visual C# Items	
▷ Web Windows Forms	Þ	EF 5.x DbContext Generator	Visual C# Items	
Workflow Xamarin.Forms	Þ	EF 6.x DbContext Generator	Visual C# Items	
Graphics Search Results	, †;	Installer Class	Visual C# Items	
▶ Online	ູ	JSON Schema File	Visual C# Items	
	្ណី	JSX File	Visual C# Items	
		MDI Parent Form	Visual C# Items	
	<u>ج</u>	OWIN Startup class	Visual C# Items	
	D	Resources File 1	Visual C# Items	
		Runtime Directives	Visual C# Items	
	e	Runtime Text Template	Visual C# Items 👻	
Name: Strings.resx 2				
				3 Add Cancel

Рис.9 Добавление ресурса с именем «Strings.resx»

После добавление появится структура ресурсного файла. В Access Modifier поменяем с Internal на Public :

Рис.10 Переключение Access modifier с Iternal на Public

В этом ресурсном файле задаются имена реализованных действий, которые будут отображаться в Robin.(см. примеры реализованных плагинов).

Проделываем теже самые шаги и для добавления ресурса для руссификации с наименованием "Strings.ru-RU.resx"

7. Добавьте в папку Actions class, например с именем вашего проекта (можно и с другим именем) он должен быть public. Поменяйте в нем namespace на Robin.RobotTools Далее в этом классе вы добавляете действия своего плагина в виде методов, которые должны быть статическими. Выглядит следующим образом:

Рис.11 Пример дефолтного представления метода с действиями плагина

8. В папку Studio добавляем классы с названиями действий плагина в нашем случае Deistvie1, Deistvie2. Проект плагина будет выглядеть следующим образом:

Рис.12 Пример дефолтного вида проетка плагина

Откроем класс Deistvie1. Добавим:

```
using Studio.Core.Base;
using Studio.Core.CustomProperties;
using System;
using System.ComponentModel.DataAnnotations;
```

3	🗉 Strir	ngs – 🏷 Add R	esource - 🗙 Remo		Access Modifier: Public -			
		Name		Value		Con	nment	
	Þ#	String1						

using System;	
namespace Robin.RobotTools	
public static class Test	
public static void Deistvie1(int Val1, string Str1) {	
<pre>public static string Deistvie2(int Val2, string Str2) {</pre>	
return string.Empty; }	



Поменяем на namespace Studio.Core.ИмяПроекта в нашем случае Studio.Core.Test. Добавим два атрибута к классу:

[Serializable] [Actions.Serialization.StudioSerialization(Guid, DisplayName)]

Guid должен быть уникальным для каждого действия!

И унаследуемся от класса abstracAction. В итоге должно выглядеть следующим образом:



Рис.13 Пример дефолтного вида действия плагина

Так как в данном примере действие(Deistvie1) принимает два параметра типа int и string добавим две переменные Str1 типа cpString и Val1 типа cpInt:





Теперь добавим к вновь добавленным переменным атрибут [Display]:

Рис.15 Добавление ammpuбута [Display]

где:

GroupName = "Common__InParameters" - ключ для группы с переменной Str1; Name = "Deistvie1_Str1_N" -ключ для наименование переменной; Order = 1 - порядковый номер отображения переключения; ResourceType=typeof(Robin.RobotTools.Properties.Strings)) - Возвращает или задает тип, содержащий ресурсы. Тут идет обращение к созданному ресурсному файлу (см. п. 7). Далее переходим к ресурсному файлу Strings.resx в Properties проекта и отредактируем его:

Рис.16 Редактирование ресурсного файла String.resx

Using Studio.Core.Base; using System; using System.ComponentModel.DataAnnotations;
Enamespace Studio Core Test
[Actions.Serialization.StudioSerialization("76990db0-63fa-4460-af1a-5020fc00e603", "Test.Deistvie1")]
🚊 class Deistvie1 : abstractAction
public override string CodeFunc => throw new NotImplementedException();
<pre></pre>
<pre>nublic cnString Strl { get: set: } = new cnString():</pre>
house choosing and (Bee) and) - way choosing())
- Dicalay/Converting _ "Common ToDanameters" Name _ "Deictvist Volt N" Orden _ 2 DecourseTune _ tunes(/Debie DebetTeals Dependence: Strings))]
[Usping(Groupwane = Commoninframenets, Name = Delstviel_vall_w, order = 2, Resource type = typeor(Robin.Robotroois.rroperties.strings))]
<pre>public cpint vall { get; set; } = new cpint();</pre>
<pre>public override string ParamNames => throw new NotImplementedException();</pre>

Name 🔺	Value	Comment
Common_InParameters	Parameters	
Common_Result	Result	
Deistvie1_Str1_N	String value	
Deistvie1_Val1_N	Int Value	

В столбце Name указываем ключи, а в столбце Value указываем названия соответственно.

Есть правила именования ключей. В вики где-то зафиксировано. Надо сослаться в то место отсюда.

Аналогично редактируется ресурсный файл для руссификации с наименованием "Strings.ru-RU. resx", только там соотвестенно используетсца кириллица.

Далее перейдем назад к классу Deistvie1 и подредактируем следующие свойства:

CodeFunc => "Test.Deistvie1" - где Test-класс где находятся действия, Deistvie1- само действие; ParamNames=>"Val1,Str1" - имена параметров.

В итоге получается:



Рис.17 Пример конечного вида действия плагина

Если действие что то возвращает, как в нашем случае: .. figure:: images/make plugin/86.png

в классе Deistvie2 проделаем те же самые действия, что и в Deistvie1, только еще унаследуемся от

интерфейса IActionReturn реализуем его и добавим using Studio.Actions; Так как Deistvie возвращает тип string, выберем из списка String:

Рис.18 Наследование от интерфейса IActionReturn

Так же добавим атрибут [Display]:

И не забудем отредактировать ресурсный файл:

public cpVar	iable R	eturnValueVa	ariable =	> new	<pre>cpVariable()</pre>	{ Vtype	e = Va	rType.	Stri	ing };	
									8	ArrayNumber	
									8	ArrayString	
									8	ArrayUnknown	
									8	Boolean	
									8	DateTime	
									8	Number	
									•	String	
- 🙆	0 Errors		2 Warnin	as	🙃 0 of 5 Mere	adec	×	Duild	•	Table	
•	U LIIUIS		.2. •••arriin	ys	U UI D WIEs:	ayes		Dulla	6	Unknown	$\overline{\mathbf{v}}$
escription					Project					File	
[Display(GroupNa public cpVariabl	me = "Com e ReturnVa	monResult", Na alueVariable =>	ame = "Commo new cpVaria	nResu	lt", Order = 100, Vtvpe = VarTvpe.9	ResourceT	ype = ty	peof(Rob	in.R	obotTools.Properties.Strin	igs))]

В итоге класс Deistvie2 должен выглядеть следующим образом:

Рис.19 Пример финального вида действия, которое возвращает результат

9. Перейдем в класс ActionLibrary.

добавим using:

```
using System.Collections.Generic;
using Studio.Core.Test;
```

Исправим namespace на Studio.Core Унаследуемся от интерфейса IActionLibrary видим следующее:

Рис.20 Наследование от интерфейса IActionLibrary

Сделаем поправки:

Рис.21 Отредактируемый класс ActionLibrary

Добавим в конструктор ActionLibrary класса наши действия и получим конечный вид нашего класса:

Рис.22 Конечный вид класса ActionLibrary

Зайдем в ресурсный файл и добавим в него Test_LB:

также добавим ключи на названия наших действий в Robin studio:

10. Перейдем в папку плагина нажмем пкм в данном примере на Test.csproj (имяПроектаПлагина.csproj) и откроем в notepad++(блокноте):

Рис.23 Открытие проекта плагина через notepad++

Строки с тэгами Configuration и Platform удаляем:

Рис.24 Удаление строк с тэгами Configuration и Platform

и вместо них вставляем следующие строки:

	Name 🔺	Value	Comment
	Common_InParameters	Parameters	
	Common_Result	Result	
	Deistvie1_Str1_N	String value	
	Deistvie1_Val1_N	Int Value	
	Deistvie2_Str2_N	String valueTwo	
Þ	Deistvie2_Val2_N	Int ValueTwo	

Ę	using	Studio.Actions;
	using	Studio.Core.Base;
	using	studio.Core.CustomProperties;
	using	system;
	using	System.ComponentModel.DataAnnotations:
	names	pace Studio.Core.Test
	{	
	ំ ខ	Serializable]
11	i i	Actions.Serialization.StudioSerialization("498a1bbb-9879-469a-a36e-56d11baa4828", "Test.Deistvie2")]
- 🗄	c c	lass Deistvie2:abstractAction.IActionReturn
		public override string CodeFunc => "Test.Deistvie2":
11		[Display(GroupName = "Common InParameters", Name = "Deistyie2 Str2 N", Order = 1, ResourceType = typeof(Robin.RobotTools,Properties,Strings))]
- 1		<pre>public costring Str2 { get: set: } = new costring():</pre>
• I I		[Display(GroupName = "Common_InParameters", Name = "Deistvie? Val2 N", Order = 2, ResourceType = typeof(Robin.RobotTools.Properties.Strings))]
- 1		<pre>public coInt Val2 { get: set: } = new coInt():</pre>
		[Display(GroupName = "Common Result", Name = "Common Result", Order = 100, ResourceType = typeof(Robin, RobotTools, Properties, Strings))]
		<pre>public covariable ReturnValueVariable => new covariable() { Vtype = VarType.Strine };</pre>
		public override string ParamNames => "Val2.Str2":
	3	
	}	
Ľ		







	Name	Value	Comment
	Common_InParameters	Parameters	
	Common_Result	Result	
	Deistvie1_Str1_N	String value	
	Deistvie2_Str2_N	String valueTwo	
	Deistvie1_Val1_N	Int Value	
	Deistvie2_Val2_N	Int ValueTwo	
•	Test_LB	Test	

	Name 🔺	Value	Comment
	Common_InParameters	Parameters	
	Common_Result	Result	
	Deistvie1_Str1_N	String value	
	Deistvie1_Val1_N	Int Value	
	Deistvie2_Str2_N	String valueTwo	
	Deistvie2_Val2_N	Int ValueTwo	
	Test_LB	Test	
	Deistvie1_N	Deistvie1	
)	Deistvie2_N	Deistvie2	
*			

Имя	<u>^</u>	Дата изменения	Тип	Размер	
Actions		22.02.2019 10:01	Папка с файлами		
bin (22.02.2019 8:30	Папка с файлами		
📔 obj		22.02.2019 10:27	Папка с файлами		
👌 Prop	erties	22.02.2019 10:43	Папка с файлами		
👩 Stud	io	22.02.2019 10:23	Папка с файлами		
🗎 Actio	onLibrary.cs	22.02.2019 10:25	Visual C# Source F	1 KB	
🔂 App.	config	22.02.2019 8:30	XML Configuratio	1 KB	
💇 Test.	csproj	<u>22.02</u> .2019 10:01	Visual C# Project f	4 КБ	
	Открыть				
	7-Zip				
	CRC SHA	•			
	Edit with Notepad++				
ĸ	Проверить на вирусы				
ĸ	Проверить репутацию в KSN				
	Открыть с помощью	+			
-	TortoiseGit)			
	Add to archive				
	Add to "Test.rar"				
	Compress and email				
	Compress to "Test.rar" and email				
	Восстановить прежнюю версию				
	Отправить)			
	Вырезать				
	Копировать				
	Создать ярлык				
	Удалить				
	Переименовать				
	Свойства				

🔚 Test.	csproj 🔀
1	<pre><?xml version="1.0" encoding="utf-8"?></pre>
2	<pre>Froject ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003"></pre>
3	<pre><import condit<="" pre="" project="\$ (MSBuildExtensionsPath) \\$ (MSBuildToolsVersion) \Microsoft.Common.props"></import></pre>
4	A spropertyGroup>
5	<pre>Configuration Condition=" '\$(Configuration)' == '' ">Debug)</pre>
6	Platform Condition=" '\$ (Platform) ' == '' ">AnyCPU
7	<projectguid>{DDBFE77C-1F7D-4B99-A55B-25E3630AF6CA}</projectguid>
8	<outputtype>Library</outputtype>
9	<rootnamespace>Robin.RobotTools</rootnamespace>
10	<assemblyname>Robin.Test</assemblyname>
11	<targetframeworkversion>v4.7.1</targetframeworkversion>
12	<filealignment>512</filealignment>
13	<autogeneratebindingredirects>true</autogeneratebindingredirects>
14	<deterministic>true</deterministic>



и вместо него вставляем следующий контент:

```
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|Win32' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <DefineConstants>TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
</PropertvGroup>
<PropertyGroup Condition=" '$(Configuration) |$(Platform)' == 'Debug|Win64' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>false</Optimize>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|Win64' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <DefineConstants>TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
</PropertyGroup>
```

В итоге файл должен иметь следующий вид:

Рис.25 Конечный вид проекта плагина

Сохраняем файл.

Зайдем в свойства нашего плагина перейдем на вкладку Build и поставим галочку XML documentation file:

11. Далее переходим в Build.sln и запускаем его:

<code>IKM</code> no Solution Build-> Add \rightarrow Existing project...:

Рис.26 Добавление нового плагина в проект Build

В открывшемся проводнике переходи заходим в папку plugins ищем плагин в нашем случае папка Test \rightarrow Test.csproj \rightarrow Открыть.

в проекте появится плагин:

Bce!!!

Запустим студию и проверим новый плагин!

Результат:

```
<?xml version="1.0" encoding="utf-8"?>
<project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
<Import Project="$ (MSBuildExtensionsPath) \$ (MSBuildToolsVersion) \Nicrosoft.Common.props" Condition="Exists ('$ ('$ (MSBuildExtensionsPath) \$ (MSBuildToolsVersion) \Nicrosoft.Common.props" Condition="Exists ('$ ('$ ('SBuildExtensionsPath) \$ ('SBuildToolsVersion) \Nicrosoft.Common.props" Condition="Exists (') \$ ('SBuildExtensionsPath) \$ ('SBuildToolsVersion) \Nicrosoft.Common.props" Condition="Exists (') \$ ('SBuildExtensionsPath) \$ ('SBuildToolsVersion) \Nicrosoft.Common.props" Condition="Exists (') \$ ('SBuildExtensionsPath) \$ ('SBuildToolsVersion) \Nicrosoft.Common.props'' Condition="Exists (') \$ ('SBuildExtensionsPath) \$ ('SBuildToolsVersion) \Nicrosoft.Common.props'' Condition="Exists (') \$ ('SBuildExtensionsPath) \$ (
          <PropertyGroup>
<BIN_PATH Condition="'$ (BIN_PATH) '== ''>$ (SolutionDit) Bin</BIN_PATH>
                 <Configuration Condition=" '$ (Configuration) ' == '' ">Debug<<Platform Condition=" '$ (Platform) ' == '' ">Win64</Platform>
                                                                                                                                                               ">Debug</Configuration>
                 <OutputPath>$ (BIN_PATH) \$ (Platform) \$ (Configuration) </OutputPath>
<ProjectGuid>{DDBFE77C-1F7D-4899-A55B-25E3630AF6CA}</ProjectGuid>
                 <OutputType>Libtary</OutputType>
<RootNamespace>Robin.RobotTools</RootNamespace>
<AssemblyName>Robin.Test</AssemblyName>
                <AssemblyName>woohn.lest</assemblyName>
<TargetFrameworkVersion><fileAlignment>S12</fileAlignment>
<AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
<Deterministic>true</Deterministic>
            </PropertyGroup>
           <PropertyGroup Condition=" '$ (Configuration) |$ (Platform) ' == 'Debug |Win32' ">
                 <PlatformTarget>AnyCPU</PlatformTarget>
<DebugSymbols>true</DebugSymbols>
                 <DebugType>full</DebugType>
                 <Optimize>false</Optimize>
                 <DefineConstants>DEBUG;TRACE</DefineConstants>
                 <ErrorReport>prompt</ErrorReport>
<WarningLevel>4</WarningLevel>
            </PropertyGroup>
          <rryopertyGroup Condition=" '$ (Configuration) |$ (Platform) ' == 'Release |Win32' ">
<PlatformTarget>AnyCPU</PlatformTarget>

                 <DerineConstantos/inves/inclantos/
<ErrorReport>prompt</ErrorReport>
<WarningLevel>4</WarningLevel>
            </PropertyGroup>
          <DebugSymbols>true</DebugSymbols>
                 <DebugType>full</DebugType>
                 <Optimize>false</Optimize>
                 <DefineConstants>DEBUG;TRACE</DefineConstants>
                 <ErrorReport>prompt</ErrorReport>
           </warningLevel>4</WarningLevel>
</PropertyGroup>
           <DebugType>pdbonly</DebugType>
<Optimise>true</Optimise>
<DefineConstants>TRACE</DefineConstants>
                 <Perineconstants/interventerinecon
<ErrorReport>prompt</ErrorReport>
<WarningLevel>4</WarningLevel>
            </PropertyGroup>
           <PropertyGroup>
<StartupObject />
           </PropertyGroup>
<ItemGroup>
                 <Reference Include="System" />
                 <Reference Include="System.ComponentModel.DataAnnotations" />
```

Application Build*	Configuration: Active (Debug)	Platform: Active (Win64)	
Build Events Debug Resources Services Settings Reference Paths Signing Code Analysis	General Conditional compilation symbols: Ø Define DEBUG constant Ø Define TRACE constant Platform target: Prefer 32-bit Allow unsafe code Ø Optimize code	Any CPU	-
	Errors and warnings Warning level: Suppress warnings: Treat warnings as errors	<u>4</u> ▼	-
	Cutput path:	\Bin\Win64\Debug\ Browse	-
	2 ML documentation file: C Register for COM interop Generate serialization assembly:	Auto]

	Записать на оптический диск	Новая папка				-
*	Имя		Дата изменения	Тип	Размер	
	📗 .git		20.02.2019 14:32	Папка с файлами		
	📙 .vs		20.02.2019 14:11	Папка с файлами		
	퉬 Bin		20.02.2019 14:14	Папка с файлами		
	icommontools 🧞		18.02.2019 8:53	Папка с файлами		
	icensetools 🦻		18.02.2019 8:53	Папка с файлами		
	👌 plugins		22.02.2019 8:30	Папка с файлами		
	🌛 PreBuild		20.02.2019 14:11	Папка с файлами		
	🌛 robot		20.02.2019 10:18	Папка с файлами		
	🌏 Scripts		20.02.2019 10:18	Папка с файлами		
≡	🛃 spy_old		20.02.2019 10:18	Папка с файлами		
	🜛 studio		20.02.2019 10:18	Папка с файлами		
	ojitignore 🧭 🥑		18.02.2019 8:52	Текстовый докум	1 KE	
	🔊 .gitlab-ci.yml		20.02.2019 10:18	Файл "YML"	2 КБ	
	🥖 .gitmodules		20.02.2019 10:18	Текстовый докум	1 KG	
	Build.sln		22.02.2019 11:17	Microsoft Visual S	53 KB	
	BuildAllProjects.ps1		18.02.2019 8:52	Сценарий Windo	8 KE	
	🚳 do_installator.bat		18.02.2019 8:54	Пакетный файл	1 KB	
	🔊 do_magic.bat		18.02.2019 8:54	Пакетный файл	1 KE	
	💋 help.chm		18.02.2019 8:52	Скомпилирован	469 KB	
	🔊 lic_bypass.bat		18.02.2019 8:52	Пакетный файл	1 KB	
	🚳 LocalBuild.bat		18.02.2019 8:52	Пакетный файл	1 KB	
	🏊 logo.jpg		18.02.2019 8:52	Рисунок JPEG	1 826 KE	
Ŧ	🔊 netstandard.dll		18.02.2019 8:52	Расширение при	96 KE	

			tion 'Build' (46 projects)			
*	Build Solution	Ctrl+Shift+B	ense			
	Rebuild Solution		Jgins			
	Clean Solution		ChatBotKafka			
	Analyze	×	Database DataConversion			
	Batch Build Configuration Manager		Dialogs			
			Email			
Ë	Manage NuGet Packages for Solution		Excel			
2	Restore NuGet Packages		FilesSystem			
	New Solution Explorer View		Properties References			
	Calculate Code Metrics		E Actions			
	Project Dependencies		Studio			
	Project Build Order		app.config			
2	Add		New Project			
₽	Set StartUp Projects		Existing Project			
っ	Undo		Existing Web Site			
1	Commit		* New Item Ctrl+Shift+A			
U	View History		* Existing Item Shift+Alt+A			
ගි	Compare with Unmodified		🚈 New Solution Folder			
Q	Blame (Annotate)		SimulateSystem			
			TableData			
X	Rename		Vision			
ç	Open Folder in File Explorer		WindowApp			
ىر	Properties	Alt+Enter	Word			
		Solution E	Explorer Team Explorer			

