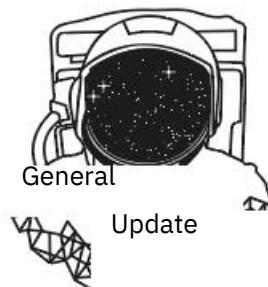


Open Source Project Author: JusterZhu (Zhu Zhen) Document Author: JusterZhu (Zhu Zhen)  
This document is provided free of charge to all  
developers in the open source community.  
Please do not use it for commercial purposes.  
Document date: 2023.1.15

---

1. Component introduction

## What is 1.1g eneralUpdate?



GeneralUpdate will be planned as a full-platform, full-stack software, with automatic update components that are not limited to any language or technology.

The open source address <https://github.com/JusterZhu/GeneralUpdate>  
~~https://gitee.com/Juster-zhu/GeneralUpdate~~ the address on the original  
github has been abandoned. Developers are requested to move to the open  
source repository address above to re-star. <https://github.com/WELL-E/GeneralUpdater>  
~~AutoUpdater Package Tool Project Address GeneralUpdate. Tools https://github.com/GeneralLibrary/GeneralUpdate.Tools~~ <https://gitee.com/GeneralTeam/GeneralUpdate.Tools> Sample project address  
<https://github.com/GeneralLibrary/GeneralUpdate-Samples> <https://gitee.com/.General Team/GeneralUpdate-Samples> Help  
Documentation Official Documentation: <http://justerzhu.cn/> Tutorial Video:  
<https://www.bilibili.com/video/BV1FT4y1Y7hV> Build the development  
environment Install the latest version of .net core SDK <https://dotnet.microsoft.com/en-us/download> Install the latest version of Visual  
Studio 2022 <https://visualstudio.microsoft.com/zh-hans/vs/getting-started/> (Visual Studio 2022 Preview Best) Get the code for this project  
<https://gitee.com/Juster-zhu/GeneralUpdate>

- 
- 
- 
- 
- 
-

Clone the code

```
//Gitee git clone https://gitee.com/Juster-zhu/  
GeneralUpdate.git
```

```
//Github git clone https://github.com/JusterZhu/  
GeneralUpdate.git
```

Branch Description master Stable Branch dev  
Development Function Branch Participation  
Contribution 1. Fork this project 2. New Feat \_ XXX  
Branch 3. Commit Code 4. New Pull Request  
Communication Free QA: <https://github.com/JusterZhu/GeneralUpdate/issues> Free Tech Exchange: 341349660

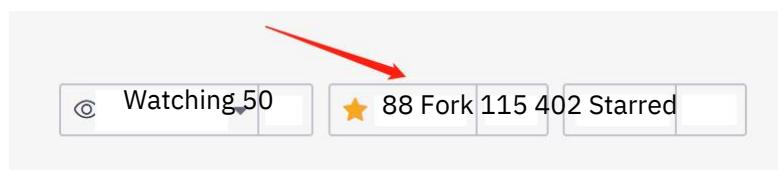
- 
- 
- Open Source Project Usage Discussions: 748744489
- Wechat Public Number:



Business cooperation, customized secondary development, technical consultation, open source project reward.

- One-to-one questions and answers about open source projects require paid consultation.
- This open source project is currently open source under the MI Open Source Agreement and is free for commercial use. In the free communication channel, we will answer the questions of developers from time to time. All community donations will be used for the development and construction of open source projects. Code contributors will be rewarded with donated funds. Mailbox: zhuzhen723723@outlook.com

A little star is your greatest support for me, so that I can keep the power of updating and maintenance.



Donation channels: 1. Open <https://gitee.com/Juster-zhu/GeneralUpdate> 2. Find the donation and click.

Fork 115 Recent Donation Record Cang K N

Donate 7 person-times, send details

C C C C C C C

3. Select the amount you support.

Juster. Zhu repository: GeneralUpdate (has received 7 donations) Open source is not easy. If it is helpful, please support it or click star. 100% of the money you donate will go to the author of the project.

Donation Amount (RMB): Donor: Juster. Zhu  
Message: Thank you for your open source project! 10 20

50 Other

Payment Type:  
Payment

Donation Cancel

## Unifying Language

Before we start using General Update, we need to know some basic concepts in the architecture.

- Client: Refers to your main application, which is the client being updated. It can also be understood as the product of the company.
- Up grade: Refers to the upgrade procedure, which will be a separate process. It needs to be placed in the same directory as the Client, and cannot be associated with any business or design in the process of use (or coding), and must be referenced independently.
- Server: The server application (ASP. NET) will provide the functions of version update information interaction and version verification.
- GeneralUpdate. Tool: is the packaging tool provided by this open source project to generate update packages (7 Z and zip).
- Update Packet: As the name implies, it is an update packet. The contents of the update packet are patches (.pacth) or new files with file content differences between the old and new versions.
- SQL: Currently, SQL scripts are generated based on Mysql database.

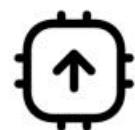
server SQL



client



upgrade



## The version number executes the standard

- Nuget Version Management Reference Standard: <https://docs.microsoft.com/zh-cn/nuget/concepts/package-versioning> Reference standard for application set Versioning: <https://docs.microsoft.com/zh-cn/dotnet/standard/assembly/versioning>
- (client programs updated by components, in general, your company's products; The operation of the component will be performed according to this standard.) In addition to updating the version number in AssemblyInfo, you also need to modify the version number of the DLL file for each iteration.

What does GeneralUpdate offer?

### Component function

Function Breakpoint resume Version by version update Binary differential update	Note Yes No Support Support Passive Effective At present, it means that the JSON configuration file with the support depth of 1 is temporarily offline. This function (passive support effective) can pop up a selection box for the user to select if it is not mandatory to update. Mandatory update directly supports the new. When a product has multiple branches, it is necessary to update the corresponding support content according to different branches. Support can verify whether the version needs to be updated. Support is implemented based on SignalR.
Incremental update function	
Configuration file Retention update Forced update Multi-branch update	
Version verification, update information	
Latest version push	

	Whether function remarks support client program, server program support C/S and B/S program can be used, application update mutual upgrade support main program can update upgrade program, upgrade program can update main program.

Packaging Tools GeneralUpdate. PacketTool desktop-side packagers written in .NET MAUI (.NET 6). Support Support Support Section

Function Differential Update	Note: Compare the previous version with the current version to find out the files that need to be updated or newly added. The generated differential update package is automatically uploaded to the server.
Package Generation Automatic	
Upload Update Package Multi-platform Support	Only Windows, Linux, MacCatalyst are supported.

What does GeneralUpdate support? .NET Framework Name .NET Core 2.0

	Yes
	No
.NET 5 6 7 8	Support
.NET Framework 4.6.1	Support

UI Framework

UI Framework Name	Support or not Support non-updatable in store mode (feedback from lindexi)
WPF	Support (currently windows only)
UWP	Support
MAUI	
Avalonia	

UI Framework NameWinUI	Support to be verified, waiting for feedback
ConsoleWinform Server	
FrameworkServer	
Framework ASP. NET	
Minimal Api OS OS Name Windows Linux	
MacCatalyst iOS Android Raspberry Pi (IoT) domestic operating system (Huawei HiSilicon, Kirin, etc.)	Yes No Support Support Support

	Supported
	Supported
	Supported
	Not supported
	Not supported
	To be verified
	To be verified

## Publish the log

[https://mp.weixin.qq.com/mp/appmsgalbum?\\_\\_biz=MzI5MTg4NzIyNg==&action=getalbum&albu\\_m\\_id=1785314218515218433&scene=173&from\\_msgid=2247483663&from\\_itemidx=nolastread=1#wechat\\_redirect](https://mp.weixin.qq.com/mp/appmsgalbum?__biz=MzI5MTg4NzIyNg==&action=getalbum&albu_m_id=1785314218515218433&scene=173&from_msgid=2247483663&from_itemidx=nolastread=1#wechat_redirect)

## 2.GeneralUpdate.AspNetCore

### Introduction

The server application (.NET Minimal Api) will provide version update information and version verification information to determine whether the update is required and the download address of the update package.

- Function

Supported Function	Remark
PushSupport	The function is based on Signal R to push the latest version update.
UpdateSupport	Check whether update is required according to the information uploaded by the client, and return the information required for update.
UploadSupport	And is use for uploading an update package GeneralUpdate. PacketTool. The implementation logic of storing the uploaded information into the database and saving the compressed package to the file server needs to be implemented by the developer, which has been reserved in the example structure.

- MySql database scripts

```
CREATE TABLE IF NOT EXISTS `updateversioninfo` ( `MD5` varchar(64) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT 'update' COMMENT 'MD5 code', `PubTime` int (11) NOT NULL DEFAULT '0' COMMENT 'version release time', `name` varchar (64) COLLATE UTF8MB4 _ Unicode _ ci DEFAULT 'version name' COMMENT 'version name', `URL` varchar (255) COLLATE utf8mb4 _ Unicode _ ci NOT NULL DEFAULT 'update URL' COMMENT 'update package file server address', `Version` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT 'last version number' COMMENT 'version number', `ClientType` int (11) NOT NULL DEFAULT '1' COMMENT 'guest', `AppSecretKey` varchar (64) COLLATE utf8mb4 _ Unicode _ ci NOT NULL COMMENT 'app Key', PRIMARY KEY (`MD5`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

## Use examples

- The nugget installation

```
NuGet\Install-Package GeneralUpdate.AspNetCore -Version 1.1.1
```

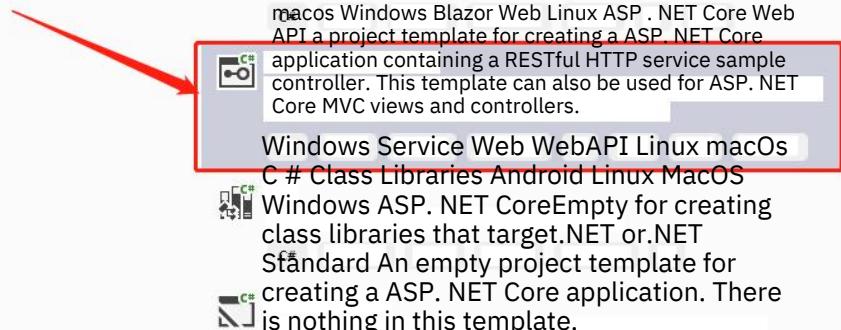
- The Minimal API is used as the service base.

Add New Project  
Recent Project  
Template Ej WPF  
Application Class  
Library

Ca

C#

Search Templates (Alt + S) (S) All Languages (L) All Platforms (P) All Project Types (T) Services Web macOS Windows Linux Blazor Server Applications Project templates for creating Blazor Server applications. The application runs on the server side inside the ASP.NET Core application and handles user interaction through the SignalR connection. This template can be used for Web applications with rich and dynamic user interfaces (UIs).  
[Next \(N\)](#) [Cloud Library](#)

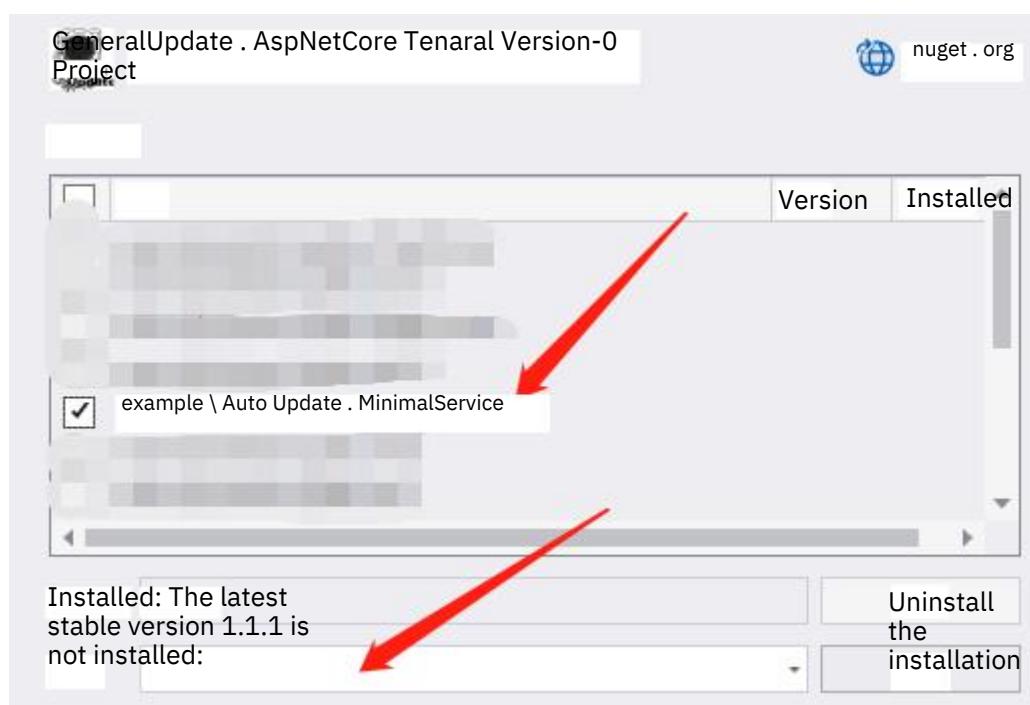
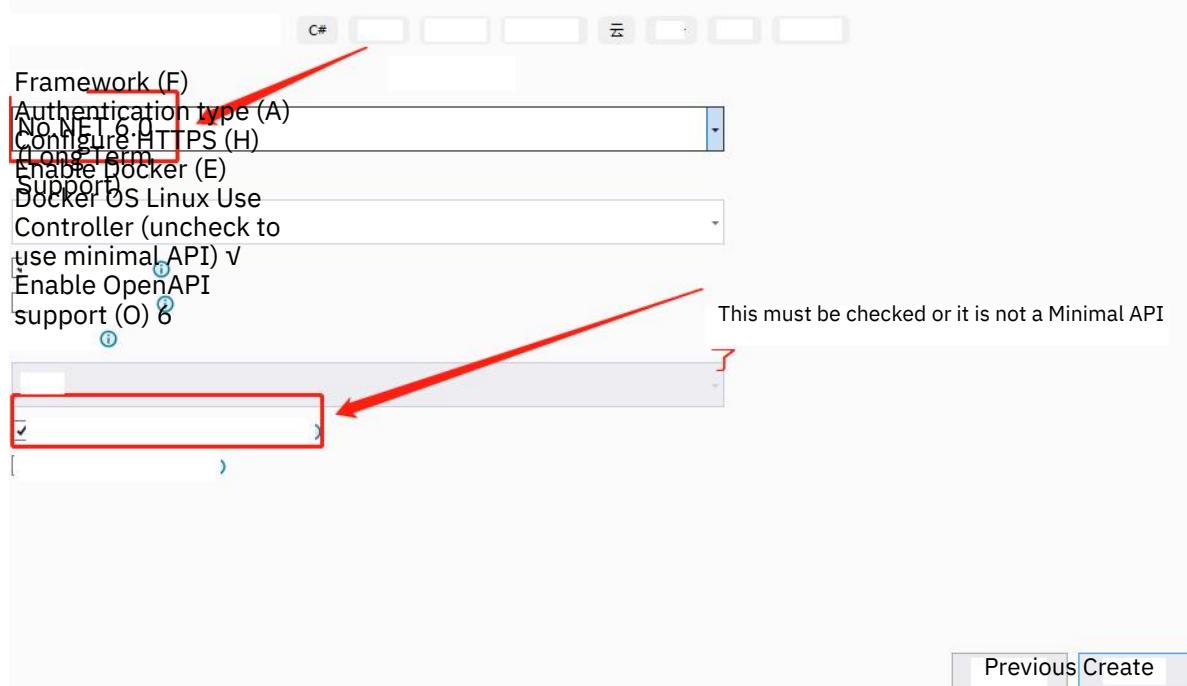


After creation, the project structure is as follows:

Configure the new project ASP .NET Core Web API Service Linux  
MacOS Windows Web WebAPI Enter a suitable name Project Name  
WebApplication1  
LocationD:\git\_community\Test Client Cloud



[Previous \(B\)](#) [Next](#)



Project structure



- Search for the GeneralUpdate . AspNetCore installation in the nuget manager.



- Sample code

Server-side complete sample code: <https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/General Update. Api/Program. CS>

Client complete sample code: <https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/General Update. Client/MainPage. XAML. CS>

My SQL scripts: [https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/sql/update\\_v2.0.sql](https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/sql/update_v2.0.sql)

### (1) Push function

- 1.1 Server-side code

```
... builder.Services.AddSignalR(); ...
```

```
/** * Push the latest version information in real time. */
app.MapHub<VersionHub>("/versionhub");
```

```
app.MapPost("/push", async Task<string> (HttpContext context) => { try
{ var hubContext =
context.RequestServices .GetRequiredService<IHubContext
<VersionHub >>(); await hubContext.SendMessage ("TESTNAME",
"123" ); } catch (Exception ex) { return ex.Message; } return "OK"; });
```

- 1.2 Client Code

```
private const string baseUrl = @"http://127.0.0.1:5001"; private
const string hubName = "versionhub";
```

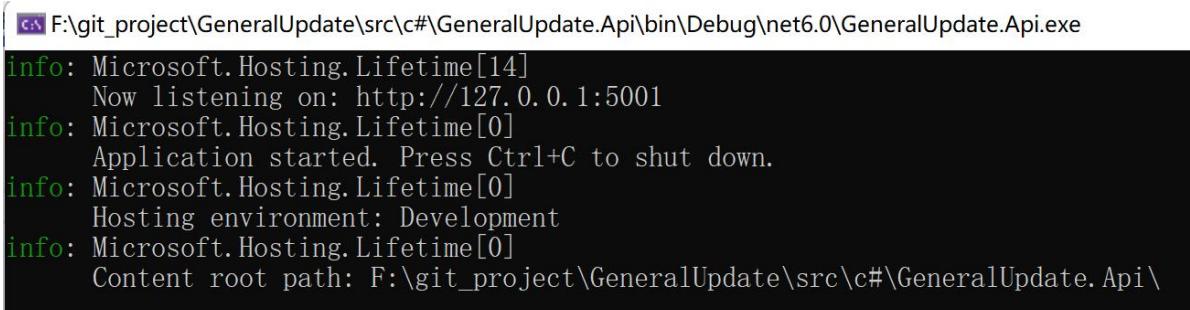
```
public MainPage () { InitializeComponent ();
MyButton.Clicked += OnClicked ; Loaded +=
OnLoaded ; }
```

```
private void OnLoaded(object sender, EventArgs e) { VersionHub
<string>.Instance.Subscribe ($"{baseUrl}/{hubName}" , "TESTNAME" , new
Action<string>(GetMessage ));
```

```
}
```

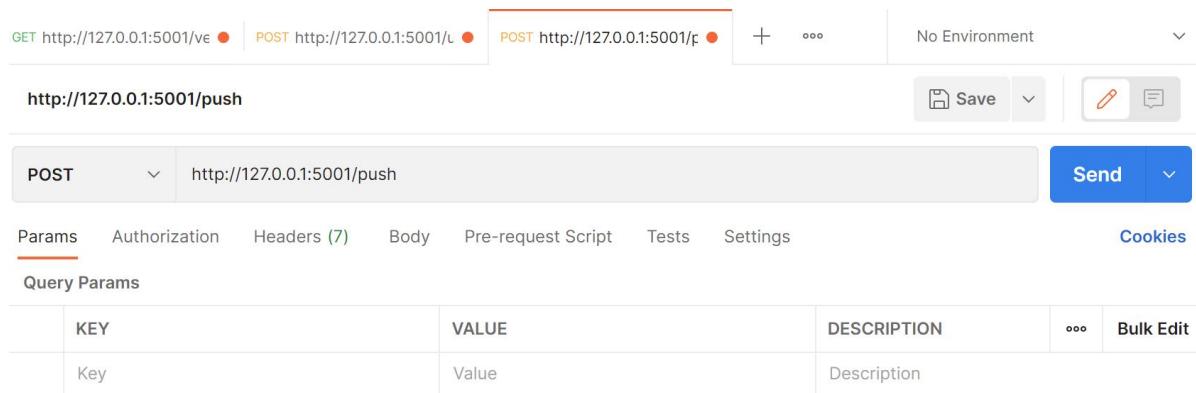
```
private async void GetMessage (string msg) { var isUpdate = await  
Shell.Current.DisplayAlert("New Version", "There are new version push  
messages !", "update" , "cancel" ); if (isUpdate) Upgrade(); }
```

- 1.3 Starting the Minimal API



```
F:\git_project\GeneralUpdate\src\c#\GeneralUpdate.Api\bin\Debug\net6.0\GeneralUpdate.Api.exe  
info: Microsoft.Hosting.Lifetime[14]  
      Now listening on: http://127.0.0.1:5001  
info: Microsoft.Hosting.Lifetime[0]  
      Application started. Press Ctrl+C to shut down.  
info: Microsoft.Hosting.Lifetime[0]  
      Hosting environment: Development  
info: Microsoft.Hosting.Lifetime[0]  
      Content root path: F:\git_project\GeneralUpdate\src\c#\GeneralUpdate.Api\
```

- 1.4 PostMan Testing

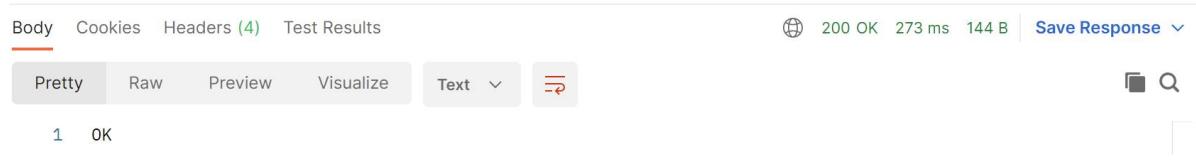


POST http://127.0.0.1:5001/push

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		



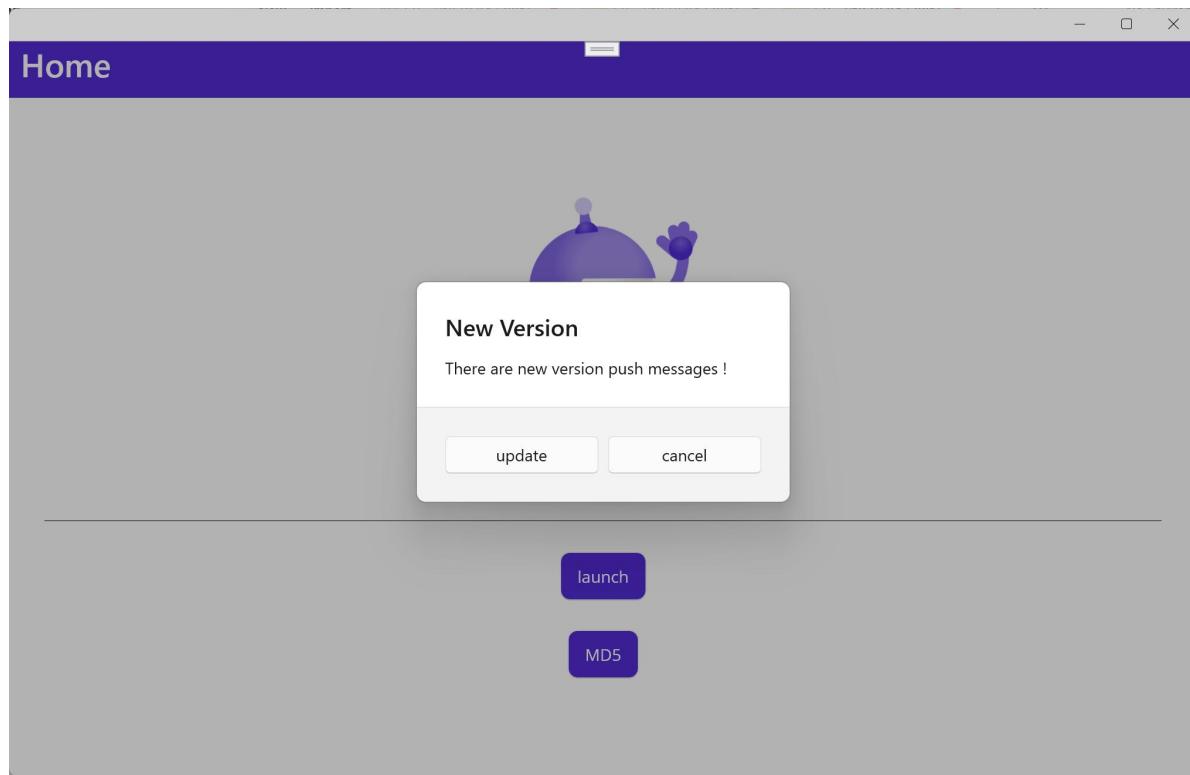
Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize Text

1 OK

Save Response

- 1.5 Operation effect



## (2) Update function

### 2.1 Server-side code

```
...  
builder.Services.AddSingleton<IUpdateService , GeneralUpdateService >(); ...
```

**/\*\* \* Check if an update is required.**

```
*/ app .MapGet("/versions/{clientType}/{clientVersion}/{clientAppKey}" , ( int  
clientType , string clientVersion, string clientAppKey, IUpdateService  
updateService) => { var versions = new List<VersionDTO>(); var md5 =  
"dd776e3a4f2028a5f61187e23089ddbd"; var pubTime = new  
DateTimeOffset(DateTime.UtcNow).ToUnixTimeSeconds(); string version =  
null ; if (clientType == AppType.ClientApp) { //client //version =  
"0.0.0.0";version = "9.9.9.9" ; } else if (clientType == AppType.UpgradeApp)  
{ //upgrad //version = "0.0.0.0"; version = "9.9.9.9" ; } var url = $"http://  
127.0.0.1/1664083126.zip" ;
```

```
var name = "1664081315"; versions.Add(new VersionDTO (md5,  
pubTime, version, url, name)); return  
updateService.Update(clientType , clientVersion, version,  
clientAppKey, GetAppSecretKey (), false , versions); };
```

Update Definition:

/// <summary> /// Verify whether the current version of the client needs to  
be updated.

/// </summary> /// <param name="clientType">1:ClientApp 2:UpdateApp</  
param> /// <param name="clientVersion">Current version of the client</  
param> /// <param name="serverLastVersion">The latest version of the server.

</param> /// <param name="clientAppkey">The appkey agreed by the client and  
server. </param> /// <param name="appSecretKey">Appkey is stored in the  
database. </param> /// <param name="isForce">Whether to force all versions to be  
updated.

</param> /// <param name="versions"></param> /// <returns>Json object.  
</returns> string Update(int clientType , string clientVersion, string  
serverLastVersion, string clientAppkey, string appSecretKey, bool isForce,  
List<VersionDTO > versions);

2.2 PostMan Testing

No Environment GET http://127.0.0.1:5001/ve POST http://127.0.0.1:5001/L POST http://127.0.0.1:5001/p ???

http://127.0.0.1:5001/versions/1/2.4.7.0/B8A7FADD-386C-46B0-B283-C9F963420C7C Save  
Send http://127.0.0.1:5001/versions/1/2.4.7.0/B8A7FADD-386C-46B0-B283-C9F963420C7C  
GET Cookies Params Authorization Body Tests Settings Headers(6) Pre-request Script Query  
Params KEY VALUE DESCRIPTION Bulk Edit ???

Key Value Description Body Cookies Headers (4) Test Results 2000K124ms401B Save  
Response Pretty Visualize Raw Preview JSON "Code":200, "Body":{ "IsUpdate": true,  
"IsForcibly":false, 00

```
1 "ClientType":1, "Versions": [ M the birthday in the  
students  
"MD5":"dd776e3a4f2028a5f61187e23089ddbd",  
"PubTime": 1665035015, "Version":"9.9.9.9",  
"Url":"http://127.0.0.1/1664083126.zip", "Name":  
"1664081315" M "Message": "Request succeeded."  
17 18  
]  
]
```

(3) Upload the update package

● 3.1 Server-side code

### /\*\* \* Upload update package.

```
*/ app.MapPost("/upload", async Task<string> (HttpContext context,  
HttpRequest request) => { var uploadReapDTO = new UploadReapDTO(); try  
{ var contextReq = context.Request; int.TryParse(contextReq .Form  
["clientType"], out int clientType ); var version = contextReq .Form  
["clientType"].ToString(); var clientAppKey = contextReq .Form  
["clientAppKey"].ToString(); var md5 = contextReq .Form["md5" ].ToString();
```

```
if (! request.HasFormContentType) throw new Exception("ContentType was not  
included in the request !"); var form = await request.ReadFormAsync(); var  
formFile = form.Files ["file"]; if (formFile is null || formFile.Length == 0) throw new  
ArgumentNullException ("Uploaded update package file not found !");
```

```
await using var stream = formFile.OpenReadStream(); byte []  
buffer = new byte [stream.Length]; stream.Read (buffer, 0,  
buffer.Length); //TODO:save to file server.
```

```
string localPath = $"E:\\"{formFile.FileName}" ; await using var  
fileStream = new FileStream(localPath, FileMode.CreateNew,  
FileAccess .Write ); fileStream .Write (buffer, 0, buffer.Length);
```

```
//TODO: data persistence.To mysql , sqlserver....
```

```
uploadReapDTO.Code = HttpStatus .OK; uploadReapDTO.Body = "Published  
successfully." ; uploadReapDTO.Message = RespMessage .RequestSucceeded ;  
return JsonConvert .SerializeObject (uploadReapDTO); } catch (Exception ex)  
{ uploadReapDTO.Code = HttpStatus .BAD_REQUEST ; uploadReapDTO.Body =  
$"Failed to publish ! Because : { ex.Message }"; uploadReapDTO.Message =  
RespMessage .RequestFailed;return JsonConvert .SerializeObject  
(uploadReapDTO); } };
```

### ● 3.2 PostMan Testing

The screenshot shows the Postman interface with a successful API call details and a table of client types.

**Request Details:**

- Method: GET http://127.0.0.1:5001/ve
- URL: POST http://127.0.0.1:5001/L+ No Environment ???
- Headers: Authorization, Headers(8), Body Pre-request Script, Tests, Settings, form-data, x-www-form-urlencoded, binary, GraphQL, none, raw, KEY VALUE
- Cookies: Headers (4)
- Body: JSON "Code":200, "Body":"Published successfully.", "Message":"Request succeeded."

**Table Data:**

version	clientType	DESCRIPTION
<input checked="" type="checkbox"/>		1
<input checked="" type="checkbox"/>	clientAppkey	1.1.1.1
<input checked="" type="checkbox"/>	md5	123456789
<input checked="" type="checkbox"/>	pretty Value	Key hfs2.3m.zip x file 123456789

**Response Summary:**

- Save Response
- 2000K54ms218B Description

### ● 3.3 Operation effect

Fill in the required parameters. Specific usage reference: The  
http://justerzhu.cn/index.php/generalupdate-packettool/  
generates an update package and publishes it to the server.

General Update

F:\temp\source Source path: F:\temp\target Target path : F:\temp\patchs Patch path: testpacket packet name: .zip format: Default encoding : 123456789

client app key : Client clienttype: http://127.0.0.1:5001/upload server url: 1.1.1.1

currnet version:

General Update

Source path : F:\temp\source F:\temp\target Target path : F:\temp\patchs Patch path : testpac packet name : Build options format : .zip Request succeeded . Default encoding : client app key : client type : http://127.0.0.1:5001/upload server url : 1.1.1.1

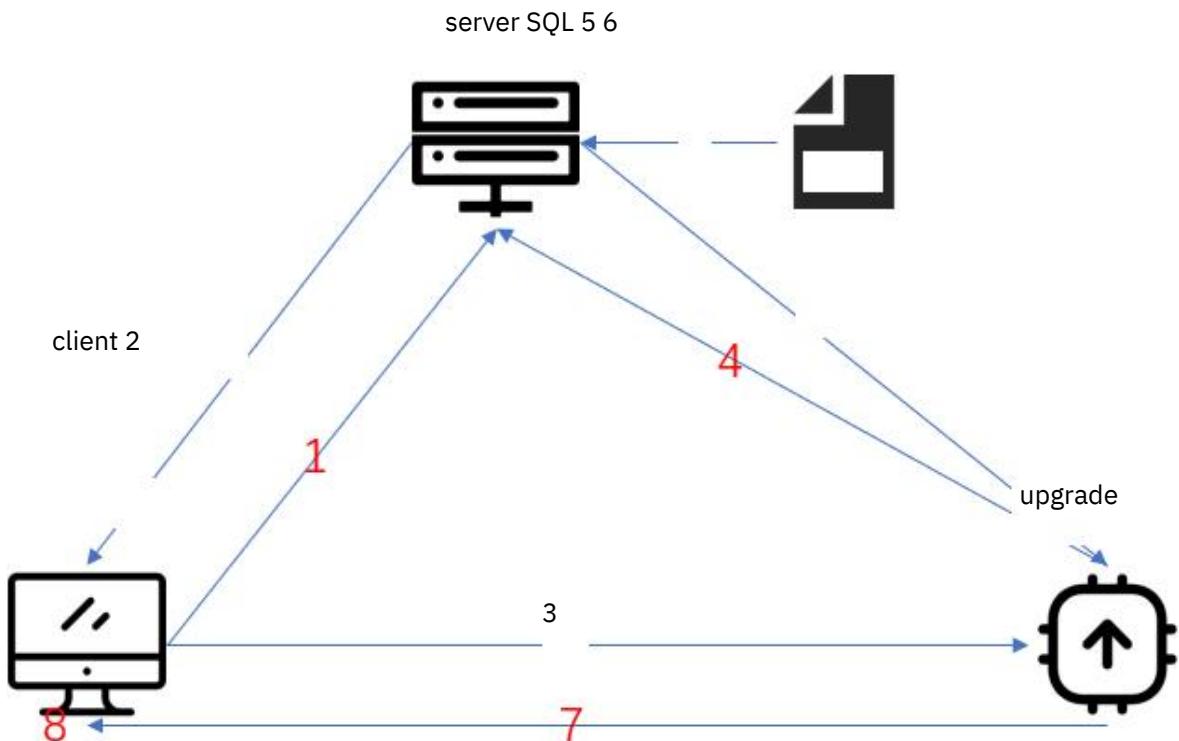
currnet version : Client 123456

ok

### 3.GeneralUpdate.ClientCore

#### Introduction

The GeneralUpdate. ClientCore is one of the core components and provides a number of key functions. There is no difference between the essence and Core, but there is a difference in responsibility. ClientCore is used in the main program. Update the upgrade assistant and then close the main program to start the upgrade assistant. (Main Responsibilities: Update and Upgrade Assistant)



The first step is to send an HTTP request to the server after the client is started to confirm whether the upgrade needs to be updated.

In the second step, if the up grade needs to be updated, the upgrade update package will be downloaded and updated.

Step 3: If the client finds that the upgrade does not need to be updated or after the upgrade is updated, it will directly start the application of the upgrade independent process through the process. (That's why you need to keep the references independent above.) Step 4: After the upgrade is started, it will automatically request the client's update package. It is used to update the content of the client. Step 5 The server will play a key role during the update of the client and upgrade requests. The version update information and the version verification information are provided to determine whether the update is required and the download address of the update package.

Step 6 After the server responds to the up grade request, upgrade will perform the operation of updating the client.

- Step 7

After the update is complete, upgrade starts the client through the process. Step 8: After the client is started, complete the update. (End of process)

Provide functionality

- Function

Features	Yes	Note
Breakpoint resume	No	Passive
Version-by-version update	Yes	Effective
Binary differential update	No	Currently, passive effective means that the JSON configuration file with a depth of 1 is supported (this function is temporarily offline and will be overwritten directly when updated). Non-mandatory update can pop up a selection box for users to choose. Mandatory update is a direct update.
Incremental update		When a product has multiple branches, Need to update the corresponding content based on SignalR implementation according to different branches
Features		
Profile retention update		
Force update		
Multi-branch update		
Latest version		

Push definition

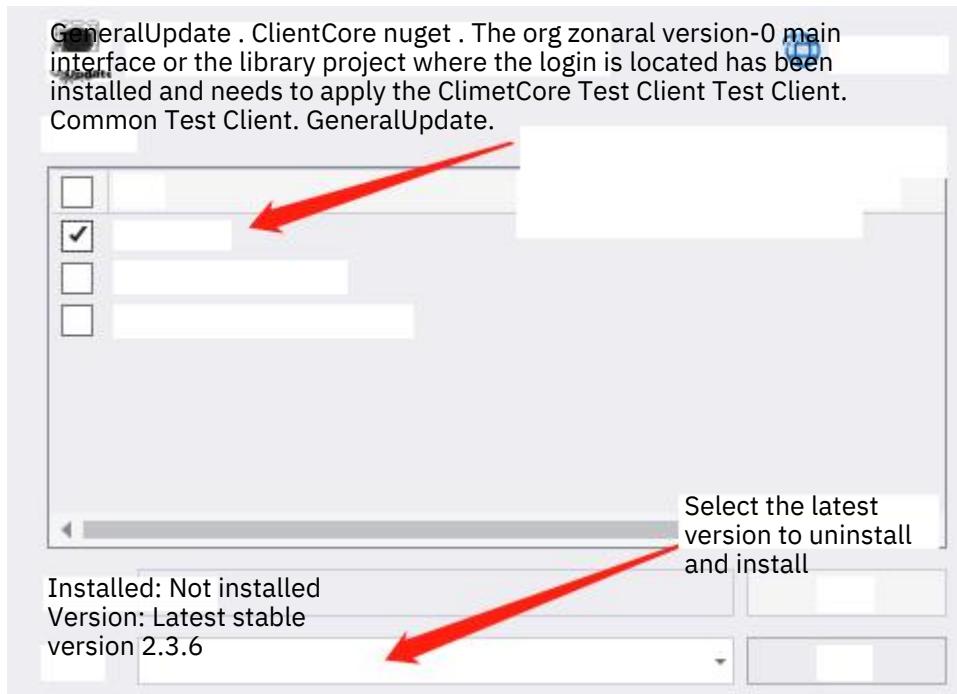
Name	General ClientBootstrap	The class notes class starts the bootstrap class.
MutiDownloadProgress Changed		Event Single or multiple update package download notification events. Events Single or multiple update package download speeds, remaining download events, and current download version information notification events.
MutiDownloadStatistics		Event Single or multiple update package download completion events. <b>Notifications of all download tasks completed.</b>
MutiDownloadCompleted		An exception notification occurred during the event download.
MutiAllDownloadCompleted		
MutiDownloadError		Any problems with the entire update process will be communicated through this event.
Exception		

Name	The class notes are used to update the relevant content configuration parameters, URL server address and port number, and the method appSecretKey client unique identifier is used to distinguish product branches.
Config	Party configuration setting method.
Option To set the download timeout configuration for the UpdateOption. Download TimeOut (in seconds).	
UpdateOption.Encoding	Enumeration sets the encoding format used for all file-related processing, such as compression, compression, and decompression.
UpdateOption.Format	Sets the compressed package format. For example, a user operation method is injected to let the user whether to skip the update under the non-mandatory update status method.
SetCustomOption	
Strategy	Set the current update strategy, for example, use WindowsStrategy only if it is a Windows platform, Linux.. mac... IOS pushes in this way.
LaunchTaskAsync	Task starts the update asynchronously. The method starts the update asynchronously.
LaunchAsync	The method automatically connect to that Signal R server and receive the latest version information of the class actively pushed by the serv.
VersionHub	
VersionHub.Instance.Subscribe	Subscriber actively pushes the latest version information (Signal R). Law

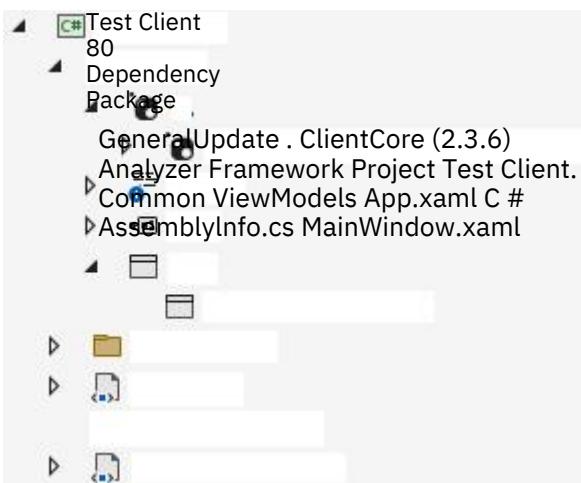
Use examples

- The nugget installation

```
NuGet\Install-Package GeneralUpdate.ClientCore -Version 2.4.7
```



After the installation is complete, you will see these in the nuget package reference.



Sample Code Complete Sample Code: [https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/GeneralUpdate.Client/MainPage.XAML.cs\(1\)Client update code](https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/GeneralUpdate.Client/MainPage.XAML.cs(1)Client update code)

```
public class MainViewModel { private const string baseUrl =  
    @"http://127.0.0.1:5001";  
  
    public MainViewModel ()  
    { Upgrade(); }  
  
    private void Upgrade() { Task.Run(async () => { var generalClientBootstrap =  
        new GeneralClientBootstrap();  
        ...  
    }); } }
```

```
//general ClientBootstrap. MultiDownloadProgress Changed of single or multiple  
update package download notification events += OnMultiDownloadProgress  
Changed; //single or multiple update package download speed, remaining download  
events, current download version information notification event general Client  
tBootstrap. MultiDownload Statistics += OnMultiDownloadStatistics; //Single or  
multiple update packages download completed general Client tBootstrap.  
MultiDownload Completed += OnMultiDownloadCompleted; //Complete all download  
tasks and notify the general Client tBootstrap. MultiAllDownload Completed +=  
OnMultiAllDownloadCompleted; //Exception notification general Client tBootstrap.  
MultiDownloadError during downloading += OnMultiDownloadError; //Any problem  
with the entire update process is notified to the general Client tBootstrap. Exception  
via this event += OnException; //Windows Strategy The update strategy will complete  
1. Automatic upgrade component self-update 2. Start Update Component 3. After  
Configinfo is configured, there is no need to write args array process communication  
as in the previous version.
```

generalClientBootstrap .Config(baseUrl, "B8A7FADD-386C-46B0-B283-C9F963420C7C").

//Set the timeout value Option (UpdateOption.  
DownloadTimeOut, 60).

//Encoding format for file processing Option (UpdateOption.  
Encoding, Encoding. Default).

//Compressed package format Option (UpdateOption.  
Format, Format. ZIP).

//Inject a func to let the user decide whether to skip this update. If it is a  
forced update, SetCustomOption (ShowCustomOption) will not take effect.

//Set the corresponding update strategy according to different platforms  
(Android Strategy, iOS Strategy, Linux Strategy, MacStrategy, Windows  
Strategy).

Have

```
Strategy<WindowsStrategy>(); #if  
MACCATALYST Strategy<MacStrategy>();  
#endif
```

```
#if ANDROID Strategy<AndroidStrategy>();  
#endif
```

```
#if IOS Strategy<iOSStrategy>(); #endif await  
generalClientBootstrap .LaunchTaskAsync()); }
```

```
private void OnMultiDownloadStatistics(object sender,  
MultiDownloadStatisticsEventArgs e) { var info = e.Version as  
VersionInfo; //info. Remaining Remaining Download Time//  
info. Speed Download Speed//info. Version Current Download  
Version Information }
```

```
private void OnMultiDownloadProgressChanged(object sender,
    MultiDownloadProgressChangedEventArgs e) { var info = e.Version as VersionInfo ; // info.TotalBytesToReceive the total size of the current update package to be downloaded/info. ProgressValue the current progress value/info. ProgressPercentage the percentage of the current progress/info. Version The currently downloaded version information.//The info. Type the currently executing action 1. ProgressType.1. Check the version information 2. ProgressType. Download is downloading the current version 3. ProgressType. Updatefile Update Current Version 4. Finished updating the ProgressType. Done. 5. Failed to update the ProgressType. Fail.//The info. BytesReceived has been downloaded.
```

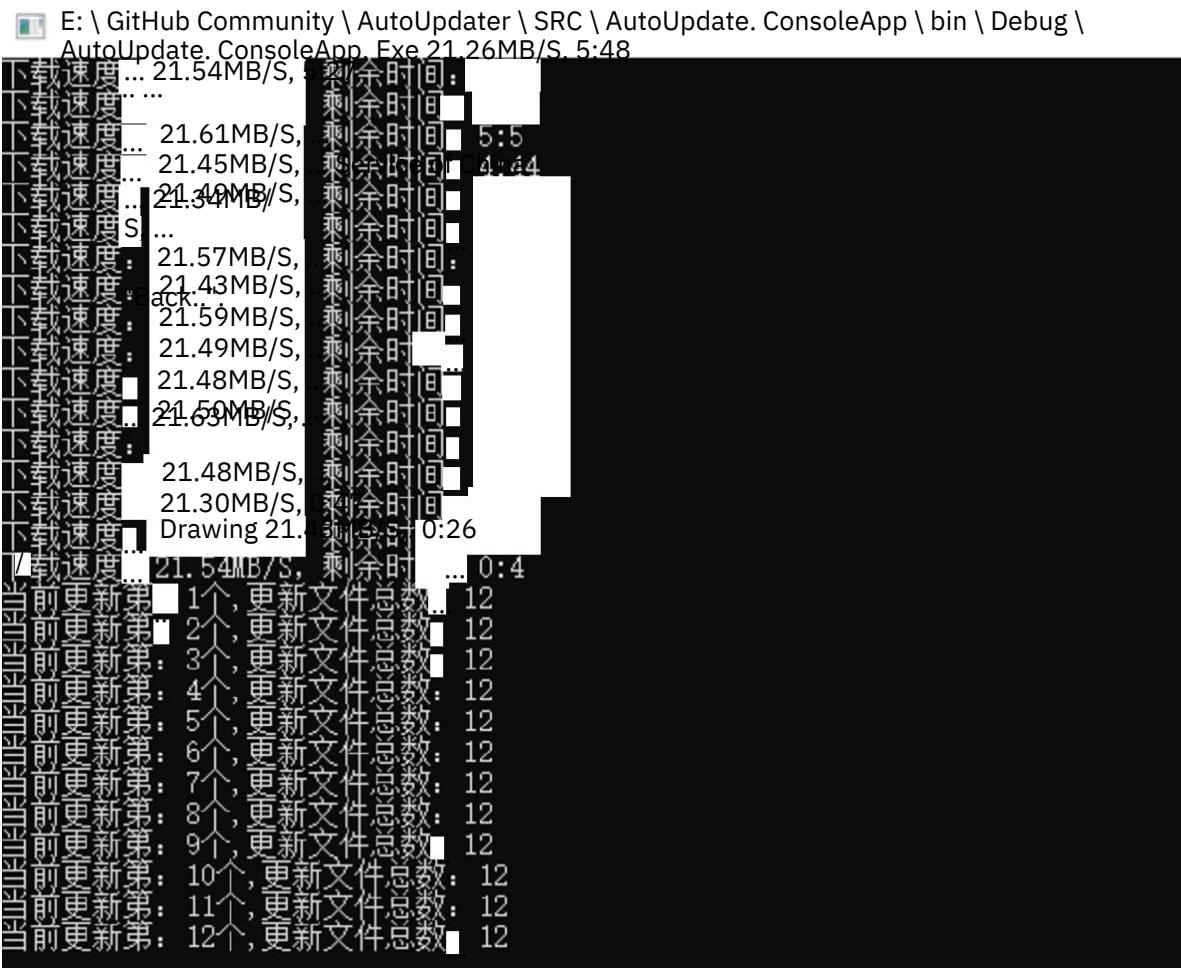
```
private void OnException (object sender, ExceptionEventArgs e)
{ Debug.WriteLine(e.Exception.Message); }
```

```
private void OnMultiAllDownloadCompleted (object sender,
    MultiAllDownloadCompletedEventArgs e) { var info = e.Version as VersionInfo ; // info.FailedVersions; If the download fails, the version of the download error and the cause of the error will be counted into the collection.
```

```
Debug.WriteLine($"Is all download completed
{ e.IsAllDownloadCompleted }."); }
```

```
private void OnMultiDownloadCompleted(object sender,
    MultiDownloadCompletedEventArgs e) { var info = e.Version as VersionInfo ; //Debug.WriteLine($"{ e.Version.Name } download completed."); }
```

```
private void OnMultiDownloadError(object sender,
    MultiDownloadErrorEventArgs e) { var info = e.Version as VersionInfo ; //Debug.WriteLine($"{ e.Version.Name } error!"); } }
```



- (2) Client subscription push code

```
private const string baseUrl = @"http://127.0.0.1:5001";  
private const string hubName = "versionhub";
```

```
public MainPage()  
{ InitializeComponent();  
MyButton.Clicked += OnClicked;  
Loaded += OnLoaded; }
```

```
Private void OnLoaded (object sender, EventArgs e) /* *  
* Method parameter * URL: server address * name:  
unique ID of client, appSecretKey value is available.
```

\* receiveMessageCallback: Receives the server's return message event.

\* onlineMessage Callback: Client online message event.

\* reconnectedCallback: Client reconnect notification event.

```
*/ VersionHub <string>.Instance.Subscribe ($"{baseUrl}/{hubName}" ,  
"TESTNAME" , new Action<string>(GetMessage)); }
```

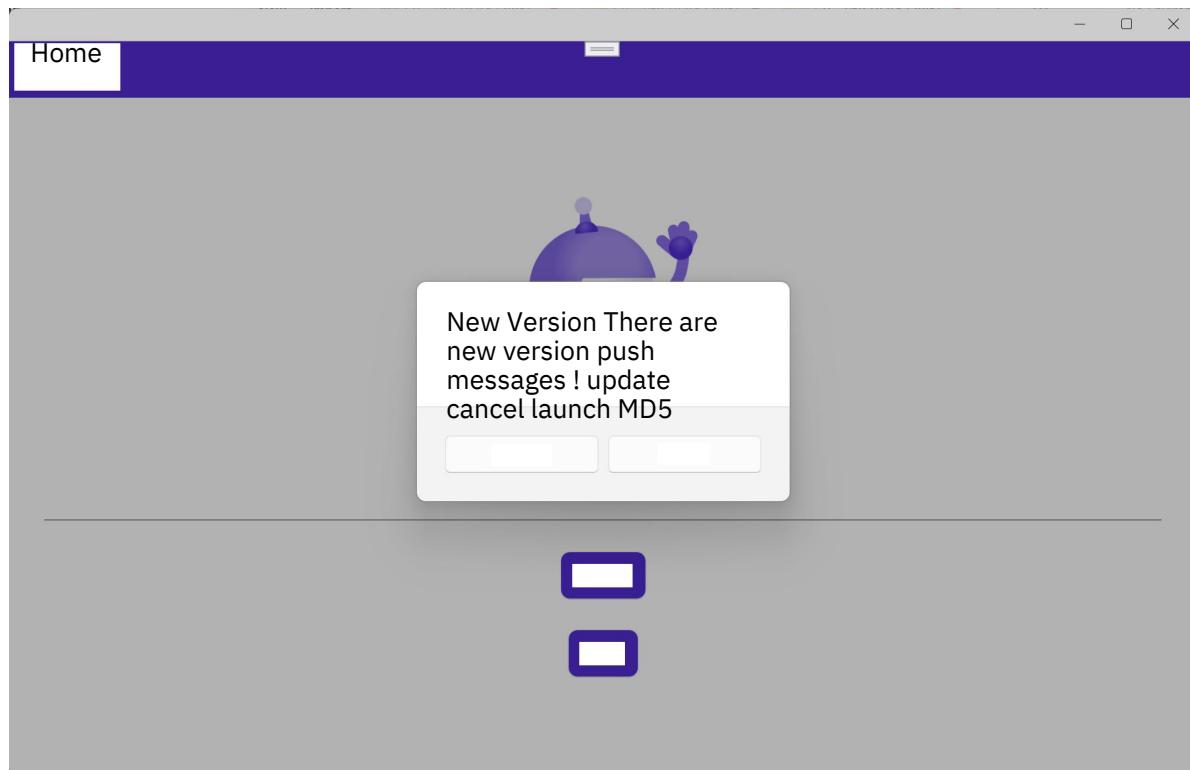
```
private async void GetMessage(string msg) {
```

```
var isUpdate = await Shell.Current.DisplayAlert("New Version", "There are new version push messages !", "update", "cancel"); if (isUpdate) Upgrade(); }
```

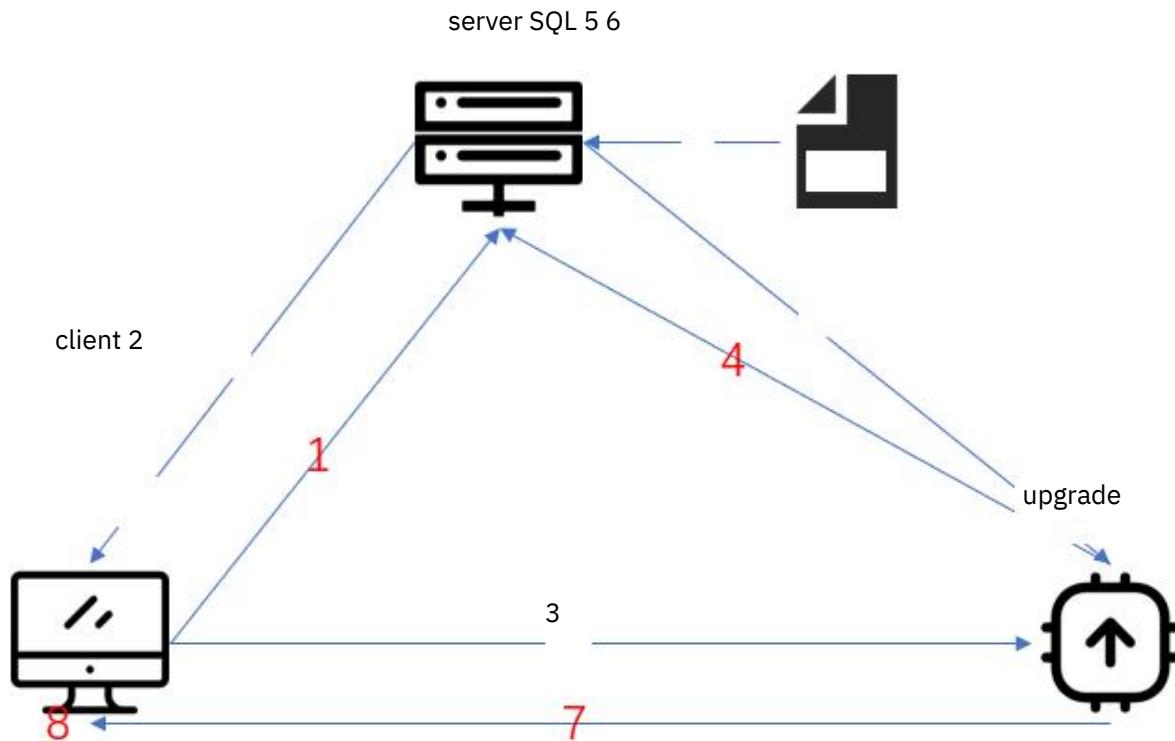
Subscribe Code Definition:

```
/// <summary> /// Subscribe to the latest version.  
/// </summary> /// <param name="url">remote server address , E.g : https://127.0.0.1:8080/versionhub . </param> /// <param name="name">The name needs to be guaranteed to be unique.  
  
</param> /// <param name="receiveMessageCallback">Receive server push callback function, The caller needs to implement the update process. </param> /// <param name="onlineMessageCallback">Receive online and offline notification callback function. </param> /// <param name="reconnectedCallback">Reconnect notification callback function. </param> /// <exception cref="Exception">Subscribe exception. </exception>  
public void Subscribe(string url, string name, Action<TParameter> receiveMessageCallback, Action<string> onlineMessageCallback = null, Action<string> reconnectedCallback = null) { //CODE.....  
  
}
```

● Running effect



The GeneralUpdate. Core is one of the core components and provides a number of key functions. After the upgrade operation of the main program is completed, the component will be called to complete the upgrade operation of the main program by starting the process and passing parameters. (Main Responsibility Update Master Program)



The first step is to send an HTTP request to the server after the client is started to confirm whether the upgrade needs to be updated.

In the second step, if the upgrade needs to be updated, the upgrade update package will be downloaded and updated.

**Step 3:** If the client finds that the upgrade does not need to be updated or after the upgrade is updated, it will directly start the application of the upgrade independent process through the process. (That's why you need to keep the references independent above.) **Step 4:** After the upgrade is started, it will automatically request the client's update package. It is used to update the content of the client. **Step 5** The server will play a key role during the update of the client and upgrade requests. The version update information and the version verification information are provided to determine whether the update is required and the download address of the update package.

**Step 6** After the server responds to the upgrade request, upgrade will perform the operation of updating the client.

**Step 7** After the update is completed, upgrade starts the client through the process.

**Step 8:** After the client is started, complete the update. (End of process)

## Provide

- Function

Feature Breakpoint resume Version-by-version update Binary differential update Incremental update Feature	Yes No Yes No	Note Passive Effective Passive Effective Currently, it means that the JSON configuration file with a depth of 1 is supported (this function is temporarily offline and will be directly overwritten when updated). Non-mandatory update can pop up a selection box for users to choose. Mandatory update is a direct update. When a product has multiple branches, the corresponding content needs to be updated according to different branches.
Profile Keep update		
Force update		
Multi-branch update		
Definition		

•

Name General ClientBootstrap	The class notes class starts the bootstrap class.
MutiDownloadProgress Changed	Event Single or multiple update package download notification events. Events Single or multiple update package download speeds, remaining download events, and current download version information notification events.
MutiDownloadStatistics	
MutiDownloadCompleted	Event Single or multiple update package download completion events. Notifications of all download tasks completed.
MutiAllDownloadCompleted	An exception notification occurred during the event download.
MutiDownloadError	Any problems with the entire update process will be communicated through this event.
Exception	
Config	Update the relevant content configuration parameters, URL server address and port number. The unique identifier of the appSecretKey client is used to distinguish the product branches. Party configuration setting method.
Option	To set the download timeout configuration for the UpdateOption. Download TimeOut (in seconds).
	Lift

Name	The class note type sets the encoding format used for all file-related processing, such as compression, compression, and decompression.
UpdateOption.Encoding	
UpdateOption.Format	Sets the compressed package format. For example, a user operation method is injected to let the user whether to skip the update under the non-mandatory update status method.
SetCustomOption	
Strategy	Set the current update strategy, for example, use WindowsStrategy only if it is a Windows platform, Linux.. mac... IOS pushes in this way.
LaunchTaskAsync	Task starts the update asynchronously. The method starts the update asynchronously.
LaunchAsync	Internally, the method component serializes the update information object into a JSON object and reorders it into base64 content for process information transfer.
Remote	

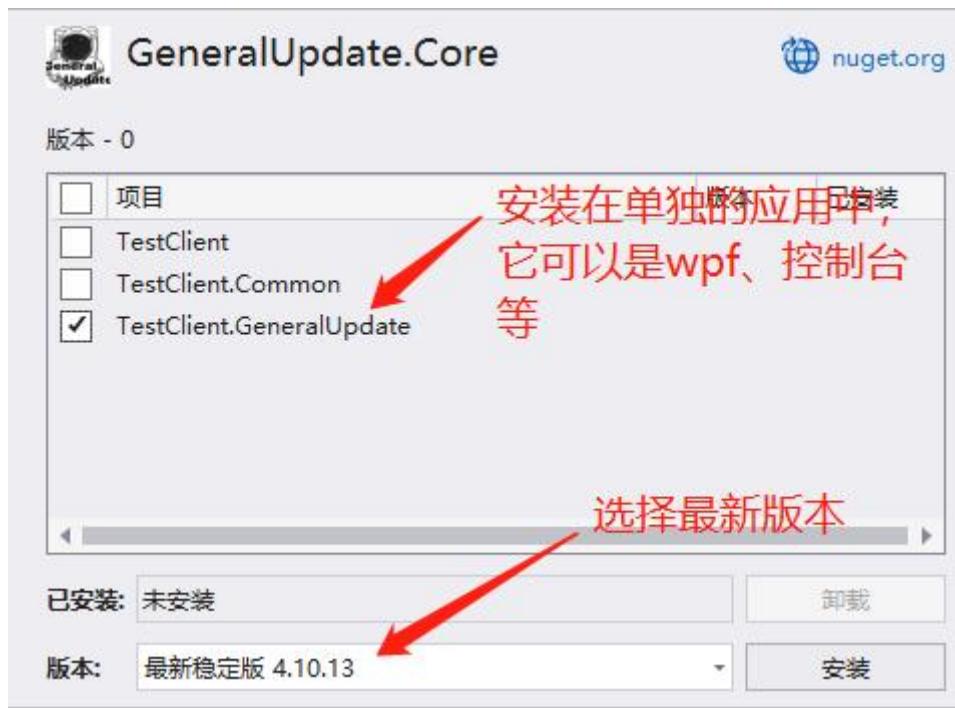
Base64 content example:

```
eyJBcHBUEXBIIjoxLCJBcHBOYW1lIjoiR2VuZXJhbFVwZGF0ZS5DbGllbnQiLCJJbnN0YWxsUGF0aCI6IkY6XFxnaXRfcHJvamVjdFxcR2VuZXJhbFVwZGF0ZVxcc3JjXFxjI1xcR2VuZXJhbFVwZGF0ZS5DbGllbnRcXGJpbLxcRGVidWdcXG5IdDYuMC13aW5kb3dzMTAuMC4xOTA0MS4wXFx3aW4xMC14NjRcXEfwcXCIsIkN1cnJlbnRWZXJzaW9uIjoiMS4wLjAuMCIsIxhc3RWZXJzaW9uIjoiOS45LjkuOSIsIkxvZ1VybCI6bnVsbCwiSXNVcGRhdGUIOmZhbHNlJCJDb21wcmVzc0VuY29kaW5nIjo3LCJDb21wcmVzc0Zvcml1dCI6bnVsbCwiRG93bmxxYWRUaW1lT3V0IjowLCJBcHBTZWNyZXRLZXkiOjCOEE3RkFERC0zODZDLTQjAtQjI4My1DOUY5NjM0MjBDN0MiLCJVcGRhdGVWZXJzaW9ucyI6W3siUHViVGltZSI6MTY2NDA5NjUyMCwiTmFtZSI6IjE2NjQwODEzMTUiLCJNRDUiOjKzDc3NmUzYTRmMjAyOGE1ZjYxMTg3ZTIzMDg5ZGRiZCIIsIlZlcnPpb24iOiIwLjAuMC4wIiwiVXJsIjoiaHR0cDovLzEyNy4wLjAuMS8xNjY0MDgzMTI2LnppcCIIsIkIEIjpudWxsfV0sIkIEIjpudWxsfQ ==
```

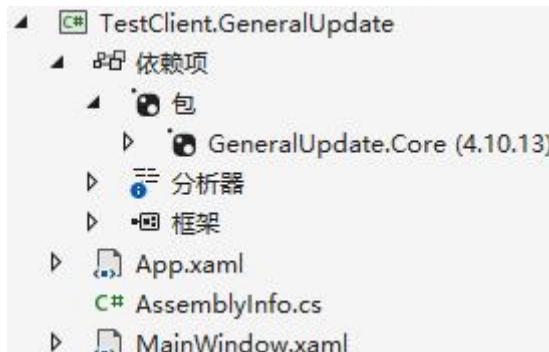
Use examples

- The nuget installation

NuGet\Install-Package GeneralUpdate.Core -Version 4.10.13



After the installation is complete, you will see these in the nuget package reference.



- Sample code

Complete Sample Code: <https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/GeneralUpdate.Upgrad/Program.cs>

- Upgrade Assistant

```
internal class Program { static void Main(string[] args) { Task.Run(async () =>
{ var generalClientBootstrap = new GeneralUpdateBootstrap(); //general
ClientBootstrap. MutiDownloadProgress Changed of single or multiple update
package download notification events += OnMutiDownload Progress
Changed; //Download speed of single or multiple update packages, remaining
download events, Current Download Version Information Notification Event
general ClientBootstrap. MutiDownload Statistics += OnMutiDownload
Statistics; //Single or multiple update packages download completed general
ClientBootstrap. MutiDownload Completed += OnMutiDownload Completed;
```

```
//Complete all download tasks and notify the general ClientBootstrap.  
MutiAllDownloadCompleted += OnMutiAllDownloadCompleted; //Exception  
notification general ClientBootstrap. MutiDownloadError during downloading +=  
OnMutiDownloadError; //Any problem in the whole update process will be notified to  
the general ClientBootstrap. Exception through this event += OnException; //Set the  
timeout value Option (UpdateOption. DownloadTimeOut, 60).
```

```
//Encoding format for file processing Option (UpdateOption.  
Encoding, Encoding. Default).
```

```
//Compressed package format Option (UpdateOption.  
Format, Format. ZIP).
```

```
//Set the corresponding update strategy according to different platforms  
(Android Strategy, iOS Strategy, Linux Strategy, MacStrategy, Windows  
Strategy). Have
```

```
Strategy < Windows Strategy > ()//The parameter passed by the main application  
obtained through process startup. The first element of the array is used by default.
```

```
.Remote(args [0]); #if MACCATALYST  
Strategy<MacStrategy>(); #endif
```

```
#if ANDROID Strategy<AndroidStrategy>();  
#endif
```

```
#if IOS Strategy<iOSStrategy>(); #endif await  
generalClientBootstrap .LaunchTaskAsync(); };
```

```
private void OnMutiDownloadStatistics(object sender,  
MutiDownloadStatisticsEventArgs e) { var info = e.Version as  
VersionInfo ; //info. Remaining Remaining Download Time//  
info. Speed Download Speed//info. Version Current Download  
Version Information }
```

```
private void OnMutiDownloadProgressChanged(object sender,  
MutiDownloadProgressChangedEventArgs e) { var info = e.Version as VersionInfo ; /  
info. TotalBytesToReceive the total size of the current update package to be  
downloaded/info. ProgressValue the current progress value/info.  
ProgressPercentage the percentage of the current progress/info. Version The  
currently downloaded version information./The info. Type the currently executing  
action 1. ProgressType.1. Check the version information 2. ProgressType.  
Donwload is downloading the current version 3. ProgressType. Updatefile Update  
Current Version 4. ProgressType. Done update completed 5. ProgressType. Fail  
update failed
```

```
//info.Bytes Received downloaded size }
```

```
private void OnException (object sender, ExceptionEventArgs e)
{ Debug.WriteLine(e.Exception.Message); }
```

private void OnMutiAllDownloadCompleted (object sender,  
MutiAllDownloadCompletedEventArgs e) { var info = e.Version as VersionInfo ; //  
info.FailedVersions; If the download fails, the version of the download error and the  
cause of the error will be counted into the collection.

```
Debug.WriteLine($"Is all download completed
{ e.IsAllDownloadCompleted }."); }
```

```
private void OnMutiDownloadCompleted(object sender,
MutiDownloadCompletedEventArgs e) { var info = e.Version as
VersionInfo ; //Debug.WriteLine($"{ e.Version.Name } download
completed."); }
```

```
private void OnMutiDownloadError(object sender,
MutiDownloadErrorEventArgs e) { var info = e.Version as
VersionInfo ; //Debug.WriteLine($"{ e.Version.Name }
error!"); } }
```

● Running effect

```

E:\GitHub社区\AutoUpdater\src\AutoUpdate.ConsoleApp\bin\Debug\AutoUpdate.ConsoleApp.exe
下载速度: 21.26MB/S, 剩余时间: 5:48
下载速度: 21.54MB/S, 剩余时间: 5:27
下载速度: 21.61MB/S, 剩余时间: 5:5
下载速度: 21.45MB/S, 剩余时间: 4:44
下载速度: 21.49MB/S, 剩余时间: 4:22
下载速度: 21.34MB/S, 剩余时间: 4:1
下载速度: 21.57MB/S, 剩余时间: 3:39
下载速度: 21.43MB/S, 剩余时间: 3:18
下载速度: 21.59MB/S, 剩余时间: 2:56
下载速度: 21.49MB/S, 剩余时间: 2:35
下载速度: 21.48MB/S, 剩余时间: 2:13
下载速度: 21.50MB/S, 剩余时间: 1:52
下载速度: 21.63MB/S, 剩余时间: 1:30
下载速度: 21.48MB/S, 剩余时间: 1:9
下载速度: 21.30MB/S, 剩余时间: 0:47
下载速度: 21.43MB/S, 剩余时间: 0:26
下载速度: 21.54MB/S, 剩余时间: 0:4
当前更新第: 1个, 更新文件总数: 12
当前更新第: 2个, 更新文件总数: 12
当前更新第: 3个, 更新文件总数: 12
当前更新第: 4个, 更新文件总数: 12
当前更新第: 5个, 更新文件总数: 12
当前更新第: 6个, 更新文件总数: 12
当前更新第: 7个, 更新文件总数: 12
当前更新第: 8个, 更新文件总数: 12
当前更新第: 9个, 更新文件总数: 12
当前更新第: 10个, 更新文件总数: 12
当前更新第: 11个, 更新文件总数: 12
当前更新第: 12个, 更新文件总数: 12

```

## 5.GeneralUpdate.Differential

### Introduction

The GeneralUpdate.ClientCore is one of the core components, providing the important functions of differential detection and incremental discovery.

### Provide functionality

Within the component, a recursive search (tree structure) will be performed to all files in the source folder directory and the target folder directory. And then traversing the MD code and the file name of each file for comparison, extracting the difference file under the two directories to generate a binary difference file and directly copying the newly added file to the update package. (Special note: the internal operation of the component will be based on the target, so the filling order of the source and target parameters cannot be reversed.) Function

- 

Functional binary differential patch new-file packaging	Yes No Yes Yes	Remark
		The difference between the old and new versions of the binary differential patch KB level. In the new version, if there are files that are not in the old version, they will be packaged directly into the update package.
	Support	

- Definition

Name	Type	Remark
Clean	Method	Clean up the files that need to be updated and pack them
Drity	Method	Update the content in the update package to the corresponding file

Source: Old version folder directory.

Target: New version folder directory.

Patches: binary differential patch, new file, etc. Generation result folder directory (compressed package in the target directory).

Use examples

- The nuget installation

NuGet\Install-Package GeneralUpdate.Differential -Version 1.0.0

- Sample code

Key Method Definitions:

Clean definition

```
/// <summary> /// Generate patch file [Cannot contain files with the same name  
but different extensions].
```

```
/// </summary> /// <param name="appPath">Previous version folder path . </  
param> /// <param name="targetPath">Recent version folder path. </param> ///  
<param name="patchPath">Store discovered incremental update files in a  
temporary directory . </param> /// <param  
name="compressProgressCallback">Incremental package generation progress  
callback function. </param> /// <param name="type">7z or zip</param> /// <param  
name="encoding">Incremental packet encoding format . </param> /// <returns></  
returns> public async Task Clean (string appPath, string targetPath , string patchPath  
= null , Action<object, BaseCompressProgressEventArgs>  
compressProgressCallback = null , OperationType type = OperationType.GZip ,  
Encoding encoding = null ,string name = null ) { //CODE...
```

}

Drity definition

```
/// <summary> /// Apply patch [Cannot contain files with the same name but different extensions] .
```

```
/// </summary> /// <param name="appPath">Client application directory .</param> /// <param name="patchPath"></param> /// <returns></returns> /// <exception cref="Exception"></exception> public async Task
```

```
Drity (string appPath, string patchPath) { //CODE...
```

```
}
```

- Different core functionality

```
//Clean up the binary difference file patch and the new file private void
BtnClean _ Click (object sender, RoutedEventArgs e) { Task.Run(async () => { var path1 = @"D:\TestCode\compare\source"; var path2 = @"D:\TestCode\compare\target"; var path3 = @"D:\TestCode\compare\patchs"; await DifferentialCore.Instance.Clean(path1, path2, path3); }); }
```

```
// "Supplement" the binary difference file patch and the newly added file to the corresponding file to complete the update of
private void BtnDrity _ Click (object sender, RoutedEventArgs e) { Task.Run(async () => { var path1 = @"D:\TestCode\compare\source"; var path3 = @"D:\TestCode\compare\patchs"; await DifferentialCore.Instance.Dirty(path1, path3); }); }
```

- Running effect

此电脑 > F (F:) > temp > patchs >			
名称	修改日期	类型	大/
temp	10/6/2022 2:06 PM	文件夹	
1.patch	10/6/2022 2:06 PM	PATCH 文件	

## 6.GeneralUpdate.Single

# Introduction

Component provides an update singleton to prevent repeated update launches. The main purpose is to avoid the problem of update confusion caused by multiple update assistants being started. (Currently, only the .NET Framework is supported, and it is the only component that cannot be cross-platform. Compatibility will be considered later.)

Provide functionality

The function prevents the update procedure from being started repeatedly.

Definition

Name	Type	Note constraint, need to inherit, implement, register, and reset the internal logic of the initial singleton object of the class that needs to be started by the singleton	Note
ISingleInstanceApp	Class		Implementation singleton
Single Instance	Class		activation
InitializeAsFirstInstance	Method		logic must
Cleanup	Method		be called
			Must be called

Use examples

- The nugget installation

NuGet\Install-Package GeneralUpdate.Single -Version 1.0.0

- Sample code

```
class Program : Application , ISingleInstanceApp { private const string AppId = "{7F280539-0814-4F9C-95BF-D2BB60023657}";
```

[STAThread] static void Main(string[] args) { //ISingleInstanceApp Can only be used in the framework of .net framework .

```
if (SingleInstance<Program>.InitializeAsFirstInstance(AppId )) { var win = new MainWindow (); var vm = new MainViewModel(resultArgs , win.Close ); win.DataContext = vm; var application = new Program(); application.Run(win); SingleInstance <Program>.Cleanup(); }}
```

```
public bool SignalExternalCommandLineArgs(IList <string> args ) { if (this.MainWindow.WindowState == WindowState .Minimized) { this.MainWindow .WindowState = WindowState .Normal;
```

```

} this .MainWindow .Activate();
return true ; } }

```

## 7.GeneralUpdate.Zip

### Introduction

The assembly provide compression and pressurization functions of that compression pack.

### Provide

#### Function Function

Compress	Yes	No		Remark
Decompress	Support			
Definition	Support			
Name	Support			
GeneralZipFactory				
UnZipProgress				
CompressProgress				
Completed				

UnZip	Type	Remarks
CreatZip	Class	The main startup class.
CreateOperate	Event	
	Event	Decompression progress.
	Event	
	Method	Compress progress.
		Operation completion notification.
		Decompress.
		Create a compressed package.
		Compression configuration.

Use examples

- The nugget installation

NuGet\Install-Package GeneralUpdate.Zip -Version 2.2.0

Sample Code Complete Sample Code: <https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/TestZIP/UnitTest1.cs>

```
/// <summary> /// Configuring  
Compression.  
/// </summary> /// <param name="type">Enumeration selects the compressed  
package format to operate on. (OperationType.GZip , OperationType.G7z)</  
param> /// <param name="name">Compressed package Name. </param> ///  
<param name="sourcePath">Source file path. </param> /// <param  
name="destinationPath">The target path. </param> /// <param  
name="includeBaseDirectory">Whether to include the root directory when packing.  
</param> /// <param name="encoding">Compressed package encoding format. </  
param> /// <returns></returns> /// <exception cref="ArgumentNullException"></  
exception> /// <exception cref="Exception"></exception> public IFactory  
CreatefOperate(OperationType type ,string name , string sourcePath , string  
destinationPath , bool includeBaseDirectory = false , Encoding encoding = null) { //  
CODE...  
}  
}
```

- Decompress

```
[Test] public void UnZip () { try { string sourcePath = "D:\\\\Updatetest_hub\\\\  
Run_app\\\\1.zip"; string destinationPath = "D:\\\\Updatetest_hub" ; string name =  
"testpacket.zip" ; var factory = new GeneralZipFactory(); factory.UnZipProgress  
+= (a,e) => { Console.WriteLine ($"fileName:{e.Name},fileSize: {e.Size},fileIndex:  
{e.Index},filePath:{e.Path},fileCount:{e.Count}" ); }; factory.Completed += (a, e)  
=> { Console.WriteLine ($"IsCompleted:{e.IsCompleted}" ); };  
factory.CreatefOperate(OperationType.GZip , name , sourcePath ,  
destinationPath , false , System.Text.Encoding.Default).
```

```
UnZip (); } catch (Exception ex)  
{ }}
```

> This computer > F (F:) >  
temp > source > name temp  
1.txt

In sour Type Size Modified Date Folder 10/6/2022  
1:48 PM 1 KB 9/23/2022 11:16 PM Text Document



To create a zip package:

```
[Test] public void CreatZip() { try { string sourcePath = "D:\\\\Updatetest_hub\\\\Run_app"; string destinationPath = "D:\\\\Updatetest_hub" ; string name = "testpacket.zip" ; var factory = new GeneralZipFactory(); factory.CompressProgress += (a,e) => { Console.WriteLine ($"fileName: {e.Name},fileSize: {e.Size},fileIndex:{e.Index},filePath:{e.Path},fileCount: {e.Count}" ); }; factory.Completed += (a,e) => { Console.WriteLine ($"IsCompleted:{ e.IsCompleted }" ); }; factory.CreatefOperate(OperationType.GZip , name , sourcePath , destinationPath ,false ,System.Text.Encoding.Default).
```

```
CreatZip (); } catch { Assert.Fail  
() ; } Assert.Pass () ; }
```



Running effect

> This computer > F (F:) >  
temp > target > name temp  
1.txt-testpacket. Zip.

Size modified at target type Date 10/6/2022 1:48 PM  
Folder 9/23/2022 11:23 PM 1 KB Text Document 2 KB  
10/6/2022 2:06 PM Compressed (zipped) Folder 0



## 8.GeneralUpdate.PacketTool

# Introduction

The tool uses .NET MAUI (.NET 6) for development and currently only supports desktop use. Helps developers to quickly generate the contents of the update package (new and old versions of the difference between the binary differential patch KB level, new files).

Provide functionality

- Function

Function	Support Note: Compare the previous version with the current version to find out the files that need to be updated or newly added. The generated differential update package is automatically uploaded to the server.
Differential update package generates automatic upload update package multi-platform support	Only Windows, Linux, MacCatalyst are supported.

- Definition

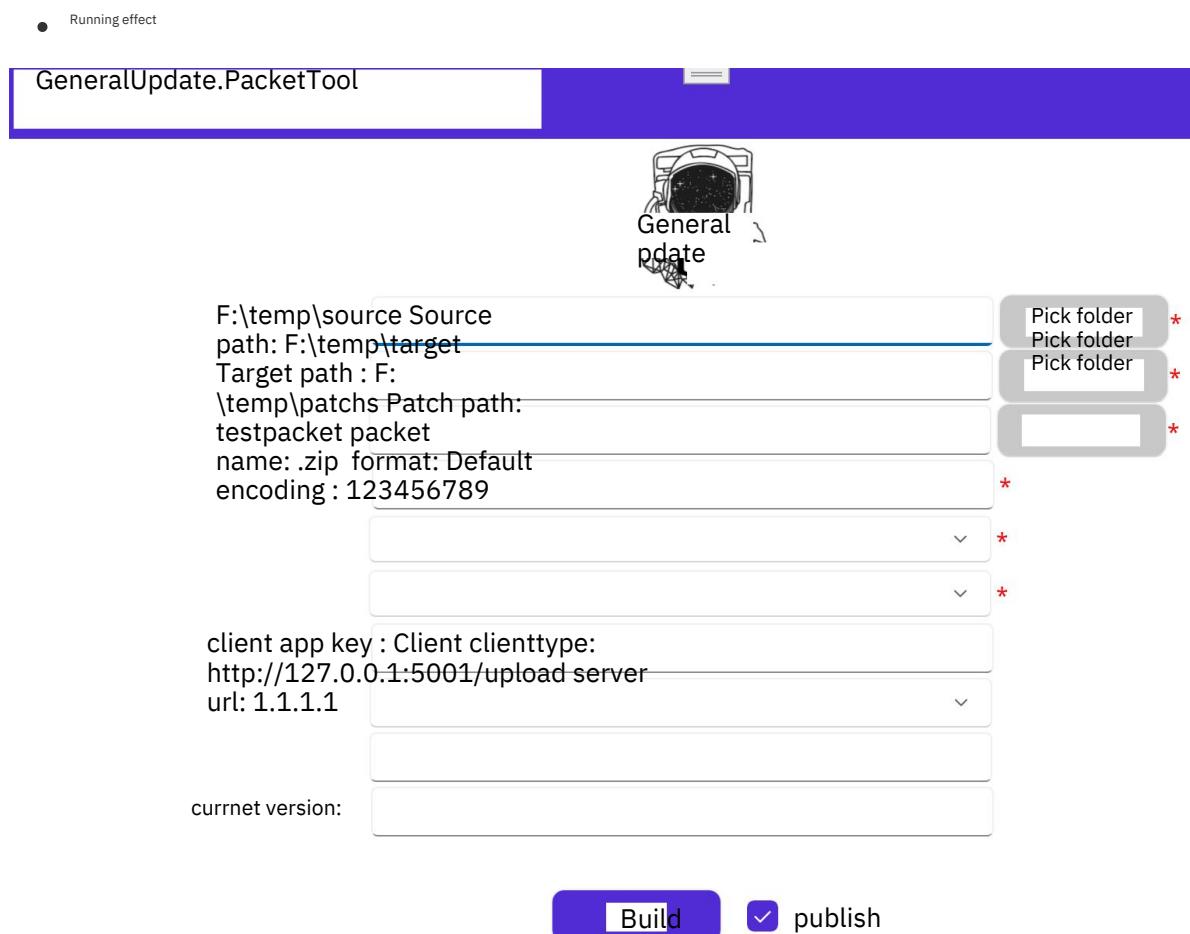
Name	Non-Comment Empty No Source path indicates the folder path of the previous version. Target path Patch path
Packet Name	No indicates the current version folder path.
Path	No indicates the path of the final generated package.
Format	No compressed package format, optional zip, 7 Z.
Encoding	Encoding is not the encoding format of the compression package. If garbled code appears, the corresponding encoding format should be specified.
Client	Client can be empty when it is not required to be uploaded and published to the server. It is used for product multi-branch marking and management. Example value: Yes app key { B13DAE4E-7881-4158-948C-24681CCA413F } Server Yes It can be blank when it is not required to upload and publish to the server. Fill in the complete interface address of the web API.
Url	
Current version	Yes, it can be left blank when there is no need to upload and publish to the server. If Upload Update Package is checked, the generated differential package will be uploaded through HTTP, and the version information of this release will be automatically inserted into the database (this operation needs to be implemented by the developer).
Publish	

Name	Non-remark empty compares all the project files (DLL, no exe..) under the Source path and Target path folders recursively. And the like), the file list needing to be updated is analyzed through the binary differential check and the incremental check, and the update package is packaged according to the folder structure. If "Publish" is checked, it will be uploaded automatically.
Build	

Use examples

Tools compile source code compile: <https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/GeneralUpdate.Tool.SLN> tools compile items with a red "asterisk", All are required. If you do not want to use the auto-publish feature, just fill in the items with a red "asterisk" (you do not need to check the "Publish" checkbox). If you need to use the automatic publishing function, you need to fill in all and check the "Publish" check box.

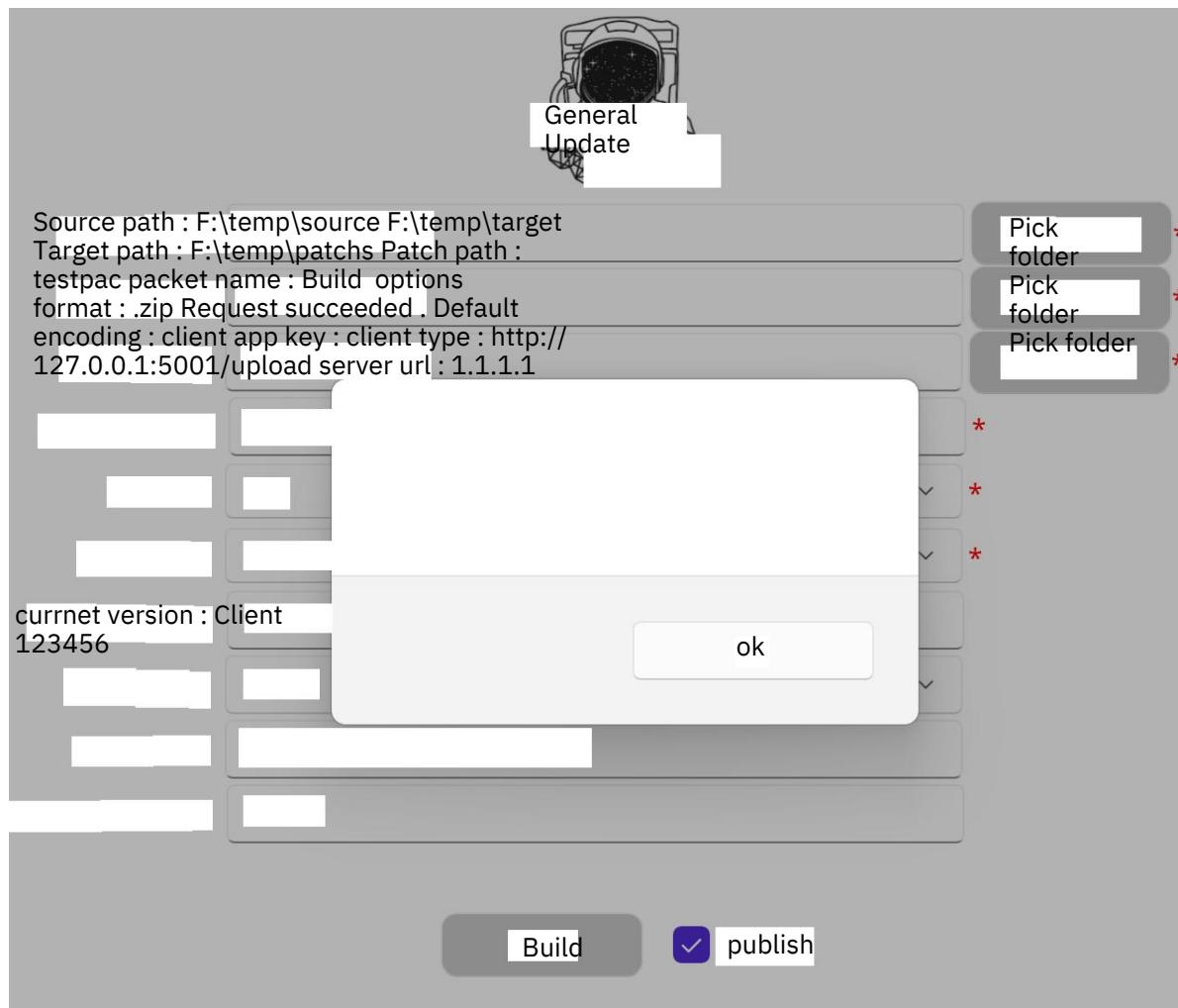
- Running effect



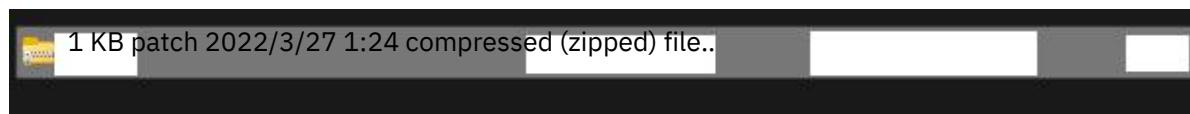
The screenshot shows the 'GeneralUpdate.PacketTool' application window. The title bar is purple. The main interface has a white background with several input fields and dropdown menus. A logo of a person wearing a mask is centered above the input fields. The fields include:

- Source path: F:\temp\source
- Target path: F:\temp\target
- Patch path: F:\temp\patchs
- Packet name: testpacket
- Format: .zip
- Default encoding: 123456789
- Client app key: (empty)
- Client clienttype: (empty)
- Upload server URL: http://127.0.0.1:5001/upload
- Current version: (empty)

On the right side, there are three 'Pick folder' buttons with red asterisks indicating they are required. Below the input fields are two dropdown menus with red asterisks. At the bottom, there is a blue 'Build' button and a checked 'publish' checkbox.



Generate Results The tool will generate the update package under the targetpath, in.zip format by default. Update the package contents



A screenshot of a table displaying patch file details. The columns include Name, Type, Compression size, Password protection, and Size. Red arrows point to the first three rows of the table:

Name	Type	Compression size	Password protection	Size
patch 321	Patch	KBKB	No	1
appsetting	PATCH	KBKB	No	1
	File Mock			
	Application			
	File Text			
	Document			
	South SON File			

## 9. Introduction to Minimal API

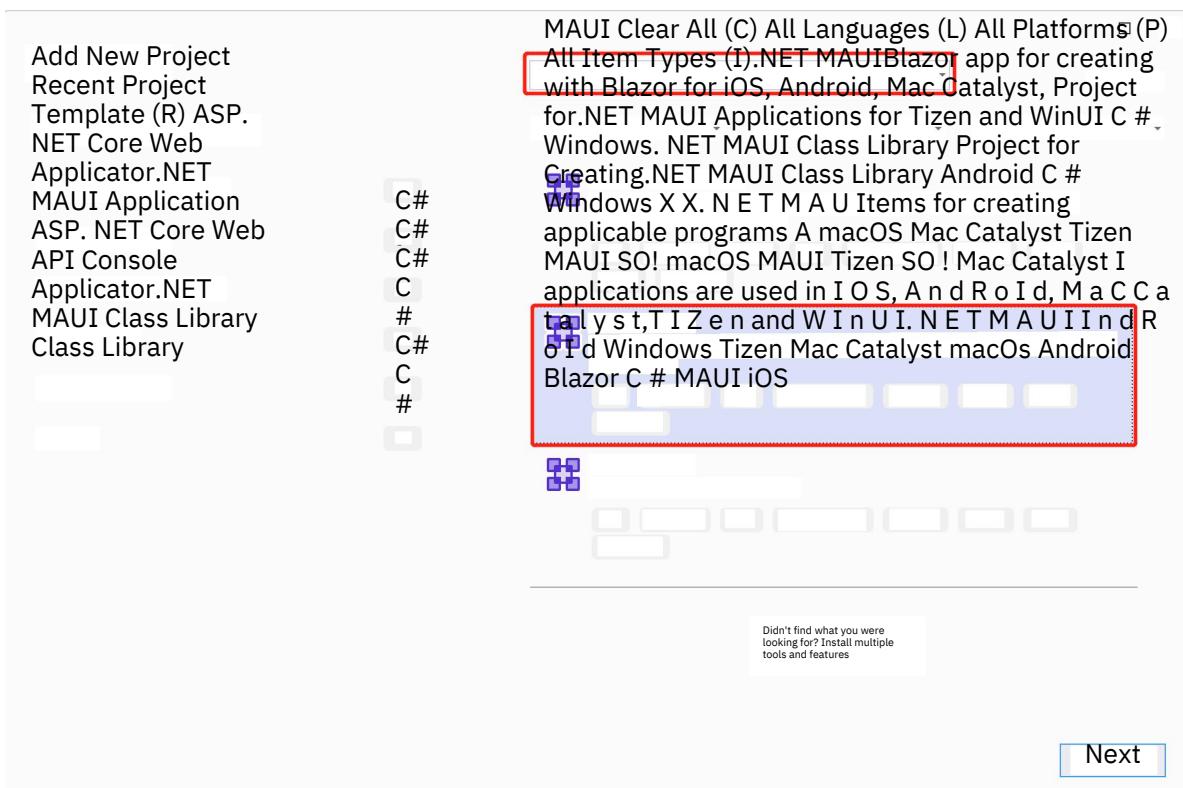
Minimal API Skills Learning: [Minimal APIs overview | Microsoft Learn] (

# 10.MAUI

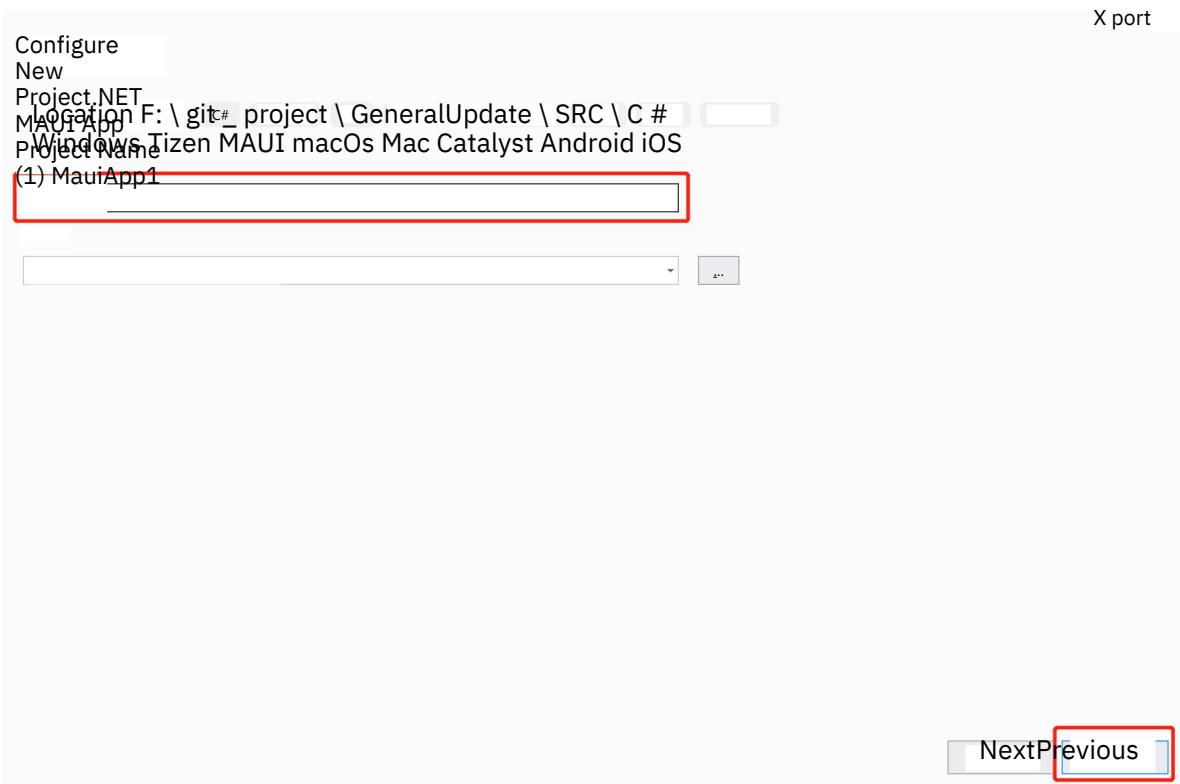
Here we share how to use GeneralUpdate in .NET MAUI. According to the current version, you need to edit the.csproj file of the .NET MAUI project. If it is not set, GeneralUpdate cannot use the process to open the client application written by .NET MAUI.

- [.NET MAUI environment configuration: UNK 1.NET MAUI Skill learning:](#)
- <https://learn.microsoft.com/zh-cn/dotnet/maui/what-is-maui>. Use example
- complete code: <https://gitee.com/Juster-zhu/GeneralUpdate/blob/master/src/c%23/GeneralUpdate.Client/MainPage.XAML.CS> Compiling the Mac Catalyst Platform Tutorial: <https://mp.weixin.qq.com/s/1Lkzd7ZxRYGbjdAPQ2xl9A>. Compiling Windows Platform Tutorial: <https://learn.microsoft.com/zh-cn/dotnet/maui/windows/deployment/overview>

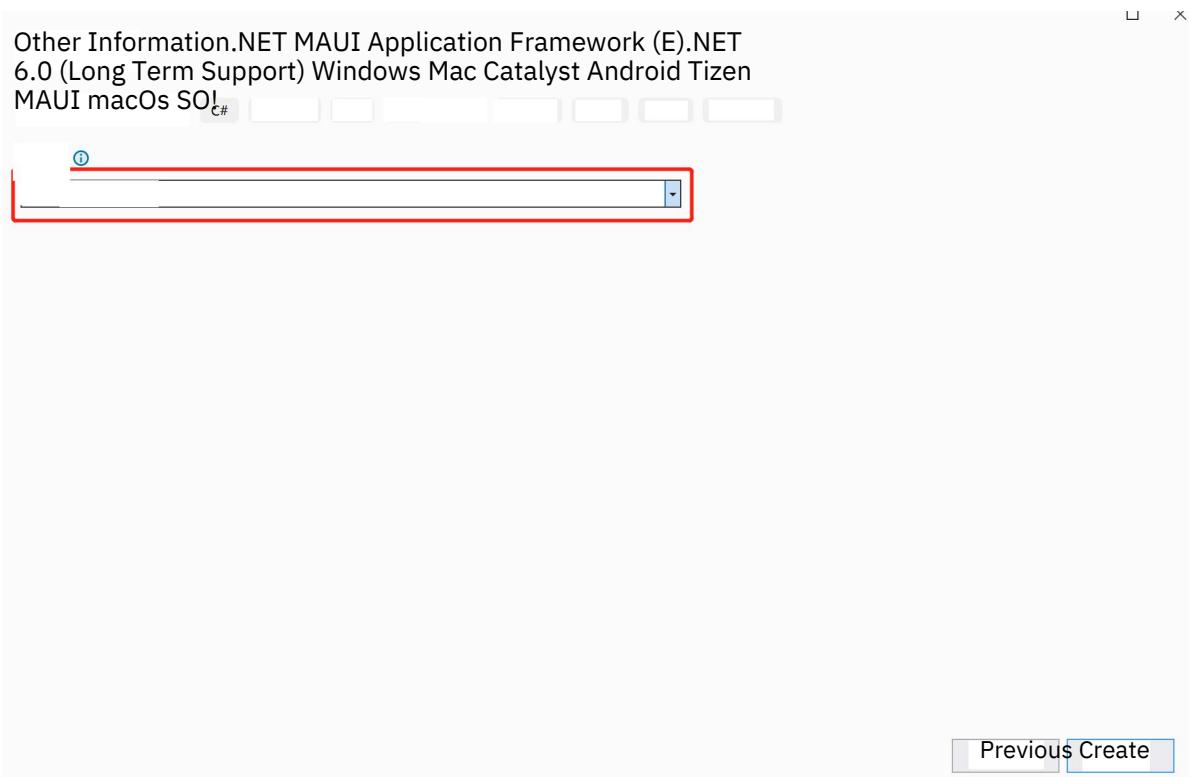
## 1. Create a new project



2. Configure the project

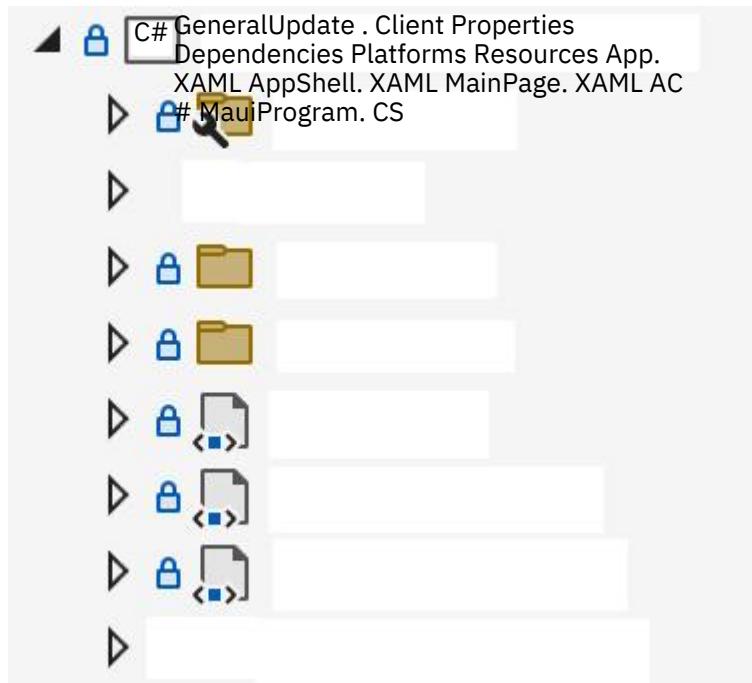


### 3. Select .NET Framework Version



### 4. Locate the .csproj file

Locate the .NET MAUI project's .csproj and double-click it directly.



## 5. Edit the.csproj file

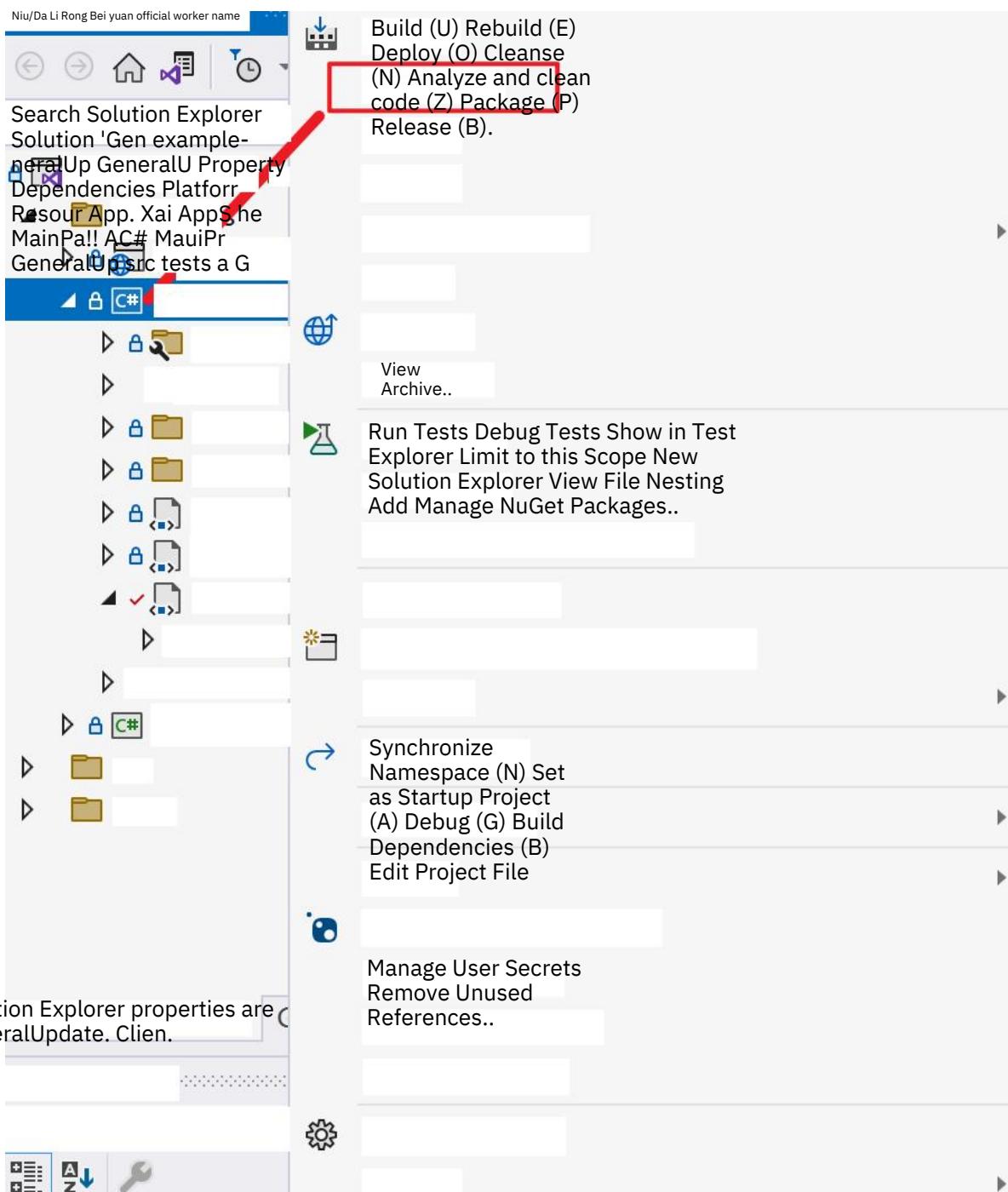
```
GeneralUpdate.Client.Csproj + day < Project Sdk = "Microsoft.NET.Sdk" > <  
PropertyGroup> < Target Frameworks > net 6.0-android; " net6.0-ios; net6.0-maccatalyst/  
TargetFrameworks> < TargetFrameworks Condition = "$([MSBuild]::IsOSPlatform('  
windows'))" $(TargetFrameworks); net6.0-windows 10.0.19041.0 /  
TargetFrameworks <!-- Uncomment to also build the tizen app. You will need to install  
tizen by following this: https://github.com/Samsung/Tizen.NET --> <!--  
<TargetFrameworks>$(TargetFrameworks);net6.0-tizen </TargetFrameworks>-->  
<OutputType>Exe </OutputType> <RootNamespace>GeneralUpdate.Client</  
RootNamespace> <UseMaui>true</UseMaui> <SingleProject>true</SingleProject>  
<ImplicitUsings>enable</ImplicitUsings> <!-- Double click to start --> <!--<  
WindowsPackageType>None </WindowsPackageType> <WindowsAppSDKSelfConatined  
> true </WindowsAppSDKSelfConatined> <!-- Display name -->  
<ApplicationTitle>GeneralUpdate.Client</ApplicationTitle> <!-- App Identifier -->  
<ApplicationId>com.companyname.generalupdate.client / ApplicationId <  
ApplicationIdGuid>087ae871-7525-4fc1-9f19-bf75ffaf73c6 / ApplicationIdGuid <!--  
Versions --> <ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>  
<ApplicationVersion>1</ApplicationVersion> <SupportedOSPlatformVersion Condition="$  
([MSBuild]: GetTargetPlatformIdentifier('$ (TargetFramework))) == ' /  
SupportedOSPlatformVersion> <SupportedOSPlatformVersion Condition="$  
([MSBuild]: :GetTargetPlatformIdentifier('$ (Targetframework)) == 'maccatalyst"-X day
```

## 6. Add tags

```
<WindowsPackageType>None </WindowsPackageType>  
<WindowsAppSDKSelfConatined>true </WindowsAppSDKSelfConatined>
```

## 7. Regenerate

"Regenerate" is required after the tag is added.

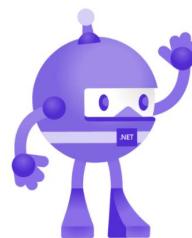
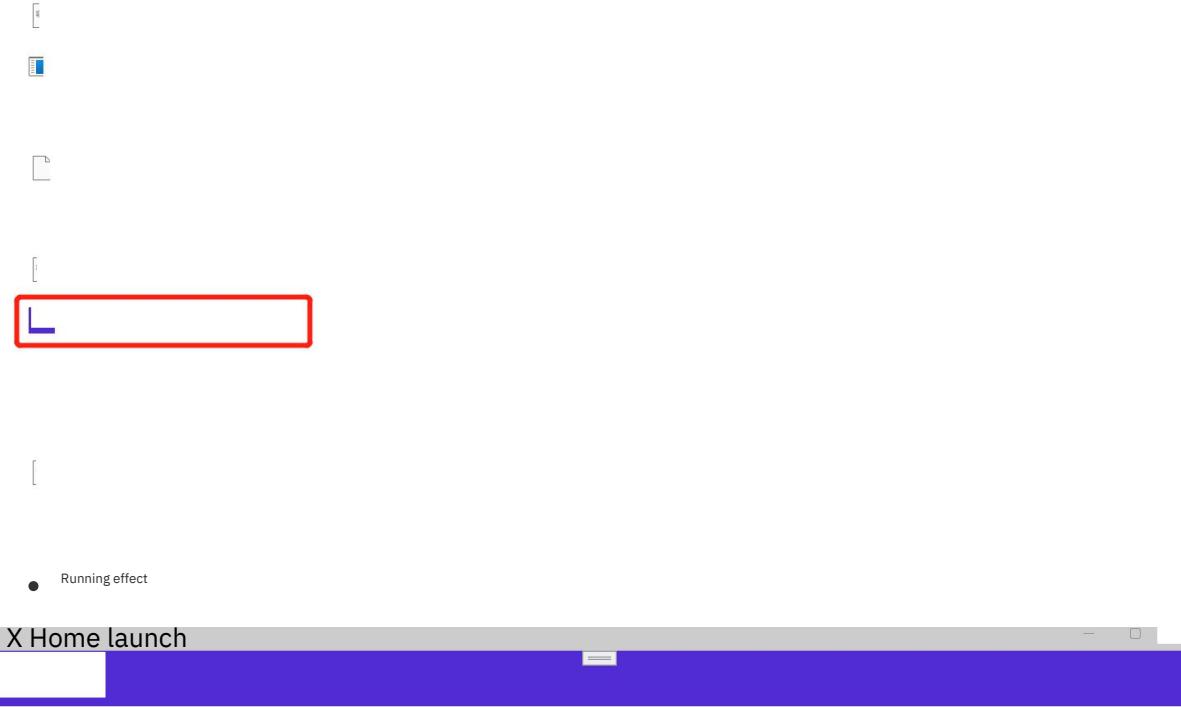


## 8. Running effect

Double-click to open or use a process to launch the .NET MAUI program that opens .exe. Reference path:

```
F:\git_project\GeneralUpdate\src\c#\GeneralUpdate.Client\bin\Debug\net6.0-windows10.0.19041.0\win10-x64
```

- Catalog



## 11. Quick start

# Use the process

Here we mainly explain how to use the General Update component from 0. The significance of this article is that you can use it after reading it in order. The code details will not be covered in this article, as they are described in other documents. By default, you have built the basic code as required.)

## 1. Make an update package

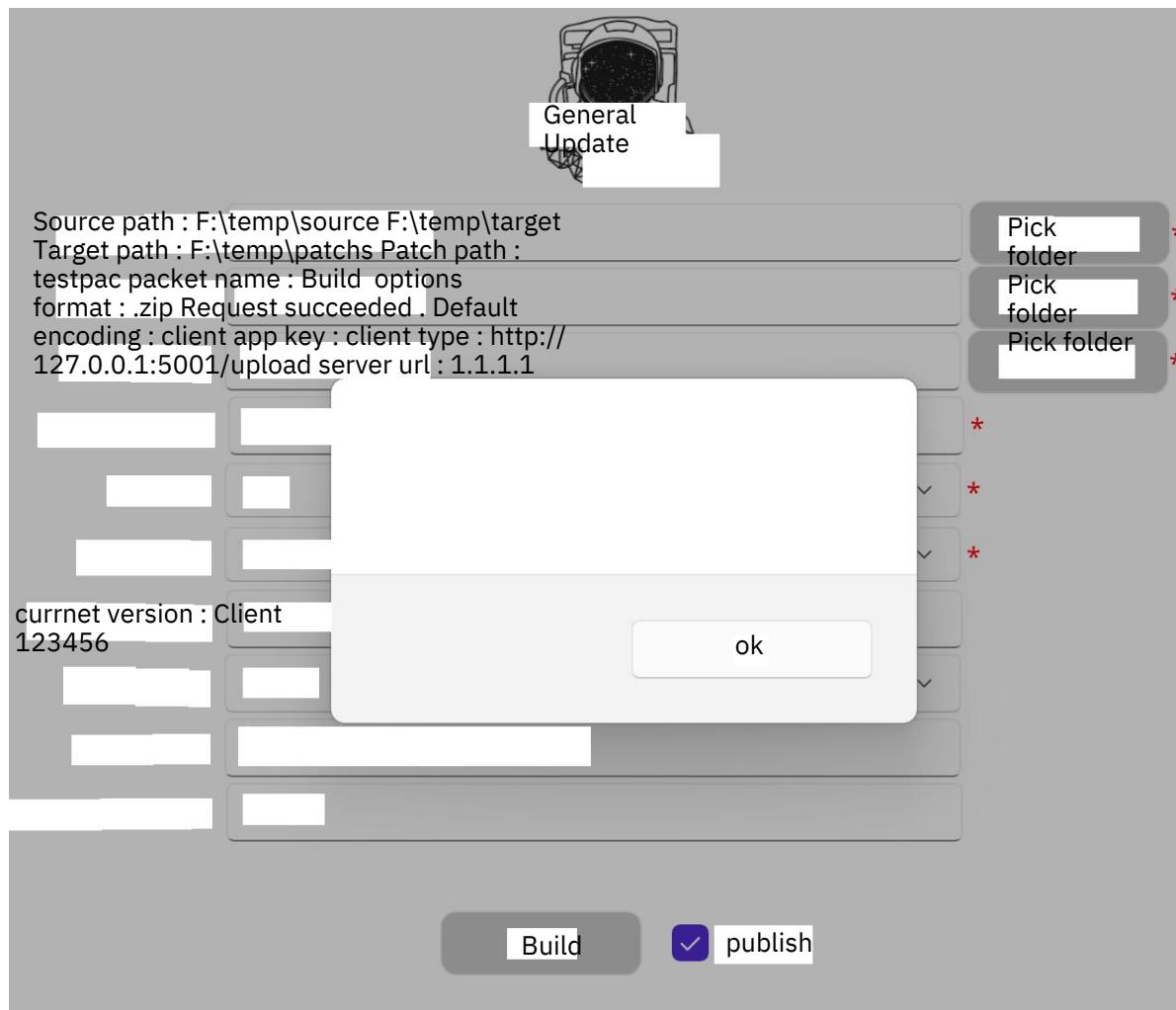
Find the PacketTool project and run it directly. (If you do not install the .NET MAUI environment, you need to install the environment according to the relevant documents provided by this site). Continue to fill in what the tool requires.

The screenshot shows the 'GeneralUpdate.PacketTool' application window. At the top, there's a logo of a satellite and the text 'General Update'. Below the logo, there are several input fields and buttons:

- Source path:** F:\temp\source Source path: F:\temp\target (highlighted in blue)
- Target path:** F:\temp\patchs Patch path: testpacket packet
- name:** .zip format: Default
- encoding:** 123456789
- client app key:** Client clienttype: http://127.0.0.1:5001/upload server
- url:** 1.1.1.1
- currnet version:** (empty field)

On the right side, there are three 'Pick folder' buttons with red asterisks indicating required fields. Below the input fields are two dropdown menus with red asterisks. At the bottom center is a purple 'Build' button, and to its right is a 'publish' button with a checked checkbox.

Click Build. If you only need to update the package, you don't need to fill in anything other than the red asterisk.



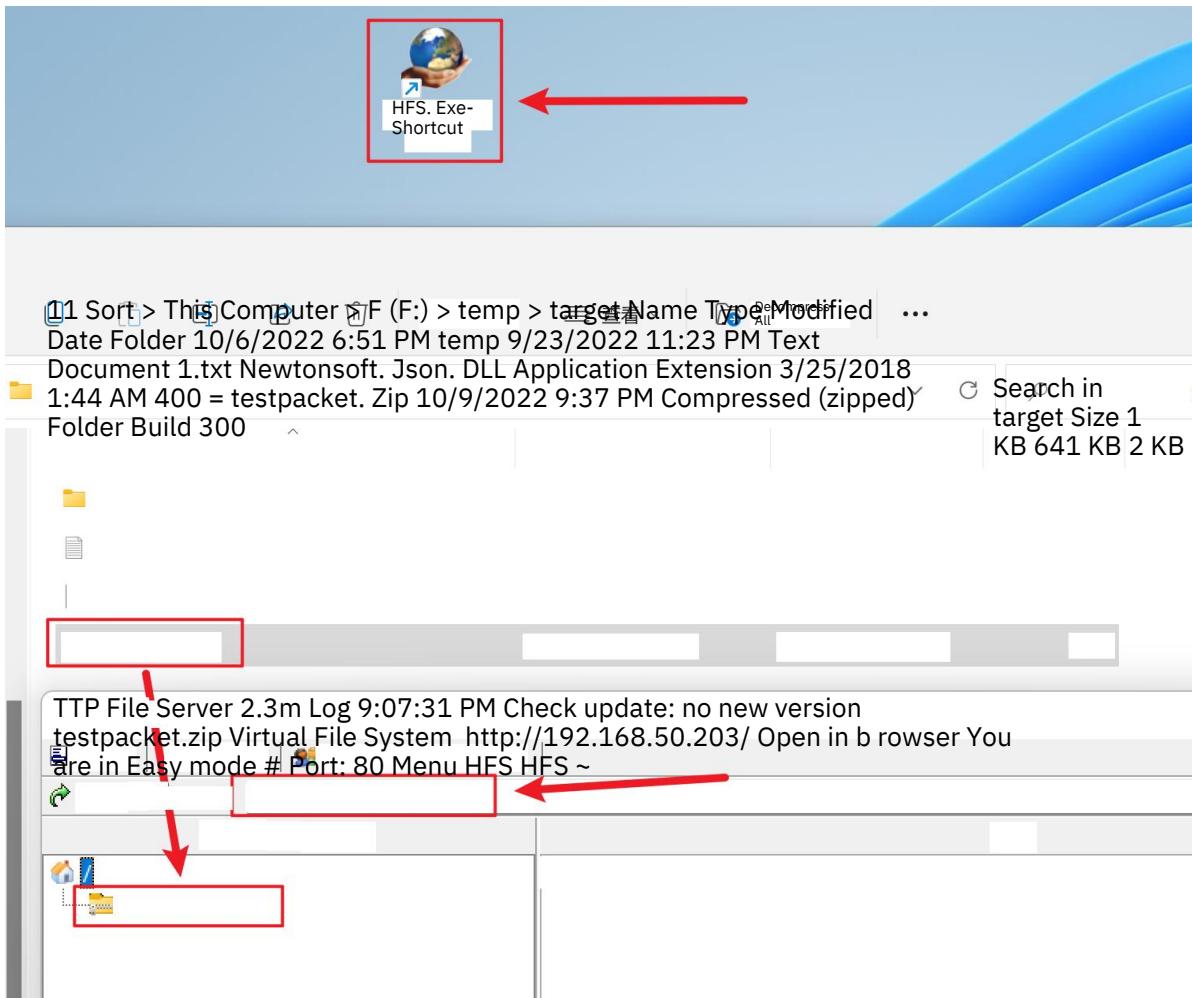
Then we get an update package.

> This computer > F (F:) > temp > target > name temp  
1.txt-testpacket.Zip. ^

Size modified at target type Date 10/6/2022 1:48 PM  
Folder 9/23/2022 11:23 PM 1 KB Text Document 2 KB  
10/6/2022 2:06 PM Compressed (zipped) Folder 0

## 2. Place the update package on the file server

Here I use the HFS tool to simulate the "file server". If it is the production environment of the company, please put it on the "file server". Find the content of the update package we have generated and drag it to the file list on the left. <Http://192.168.50.203> that this address is the "file server" address, can be modified to the address you need.



Once the file is selected, the URL of the HFS tool changes. This URL is the address we will provide to the client to download.

**HFS ~ HTTP File Server 2.3m** HFS You are in Easy mode Port : 80



Next, you need to obtain the MD5 code of the testpacket. Zip. If you check the publish partner and complete all the basic code construction on the server side to write the information into the database, you can view the MD5 code of the file in the database. Or obtain it directly through the code of SRC/C #/TestMD5/UnitTest1.cs · Juster. Zhu/GeneralUpdate- Gitee. Com, as shown below:

```
"b03d52c279faf003965c46041f2037f9"
```

Obtaining MD is used to fake data without building a complete server-side code. If there is a complete server-side build, it can be skipped.

### 3. Server-side settings

The example code on the server side is GeneralUpdate. Api. At this time, we find the code:

```
/** * Check if an update is required.

*/ app .MapGet("/versions/{clientType}/{clientVersion}/{clientAppKey}" , ( int
clientType , string clientVersion, string clientAppKey, IUpdateService updateService)
=> { var versions = new List<VersionDTO>(); var md5 =
"b03d52c279faf003965c46041f2037f9"; //The generated MD5 code of the update
package file, because when it is returned to the client, it needs to verify whether it is
available with this. Var pubTime = new DateTimeOffset (DateTime.
UtcNow).ToUnixTimeSeconds (); string version = null ; if (clientType ==
AppType.ClientApp) { //client //version = "0.0.0.0"; Version = "9.9.9.9"; //Set to 9 here
to make the program think it needs to be updated } else if (clientType == AppType.
Up gradeApp) {/up grad/version = "0.0.0.0"; version = "9.9.9.9"; //Set to 9 here to
make the program think it needs to be updated. } var URL = $ "http://192.168.50.203/
testpacket.zip"; //The download address of the update package is var name =
"testpacket";versions.Add(new VersionDTO (md5, pubTime, version, url, name));
return updateService.Update(clientType , clientVersion, version, clientAppKey,
GetAppSecretKey (), false , versions); });


```

Then assign the MD5 we just got, and fake the data according to the above code for other contents.

```
var md5 = "b03d52c279faf003965c46041f2037f9";
```

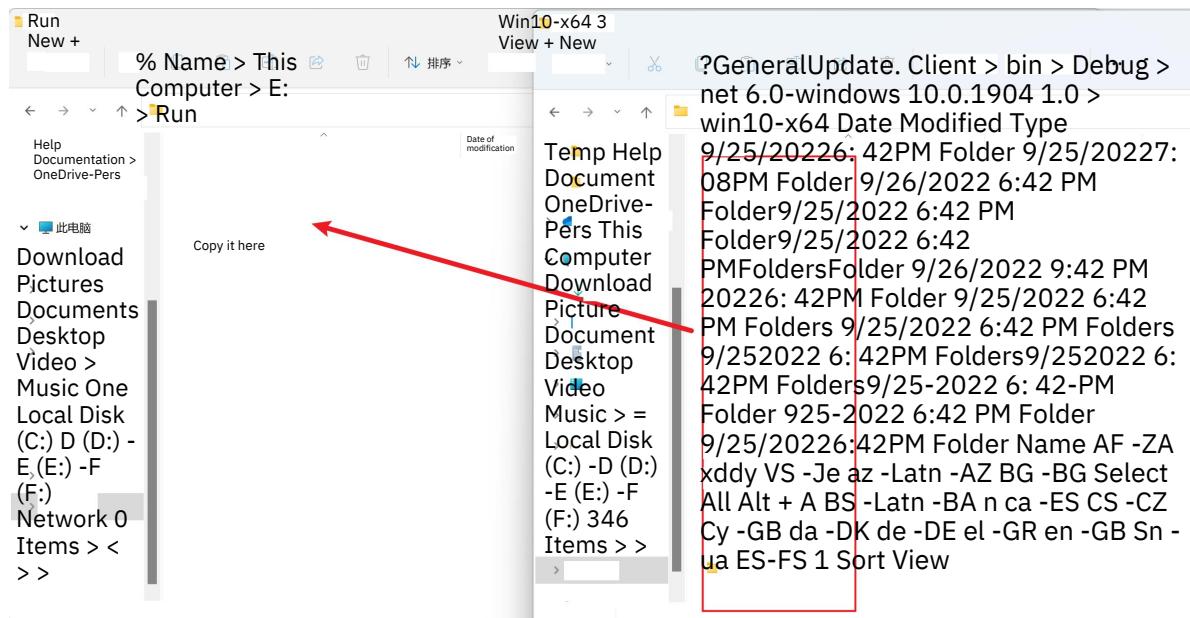
### 4. Client settings

- > This Computer
- > E (E:) Name Run



First create an empty folder Run, and copy the file list in the Bin directory of the GeneralUpdate. Client and GeneralUpdate. Upgrad to the Run directory. Before copying, you need to check whether the GeneralUpdate. Client is labeled. Reference:

[.NET MAUI – GeneralUpdate Doc \(justerzhu.cn\)](https://justerzhu.cn) .

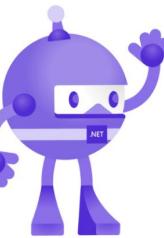


After the copy is complete, it looks like this.

▶ This Computer > E (E:) > Run Name Date Modified  
 GeneralUpdate.Client.DLL 10/9/2022 10:16 PM  
 10/9/2022 10:16 PM GeneralUpdate.Client.Exe.NET  
 GeneralUpdate.Client.pdb 10/9/2022 10:16 PM J  
 GeneralUpdate.Client.runtimeconfig.json 10/9/2022  
 10:16 PM GeneralUpdate.ClientCore.dll 10/9/2022  
 10:16 PM 10/9/2022 10:16 PM S  
 GeneralUpdate.ClientCore.pdb GeneralUpdate.Core.  
 DLL 9/25/2022 6:41 PM Product GeneralUpdate.Core.  
 PDB 9/25/2022 6:41 PM GeneralUpdate.Differential.  
 DLL 9/25/2022 6:41 PM  
 GeneralUpdate.Differential.pdb 9/25/2022 6:41 PM J  
 GeneralUpdate.Upgrad.deps.json 9/25/2022 6:41 PM  
 GeneralUpdate.Upgrad.dll 9/25/2022 6:41 PM  
 9/25/2022 6:41 PM GeneralUpdate.Upgrad.Exe  
 GeneralUpdate.Upgrad.PDB 9/25/2022 6:41 PM J  
 GeneralUpdate.Upgrad.Runtimeconfig.JSON  
 9/25/2022 6:41 PM Product

	O in Run Size
Type	238 KB 299
Application	KB 34 KB 1
Extension	KB 179 KB
Application	86 KB 101KB
Program Debug	40 KB 89 KB
Da.. Json File	49 KB 34 KB
application	10 KB 145
extension	KB 12 KB 1
Program Debug	KB
Da.. Application	
extension	
Program Debug	
Da.. Application	
extension	
Program Debug	
Da.. The JSON	
File application	
extends the	
application	
Program Debug	
Da.. JSON File	

Next, double-click the GeneralUpdate.Client.Exe to run, and then click launch (this is the end of normal operation).

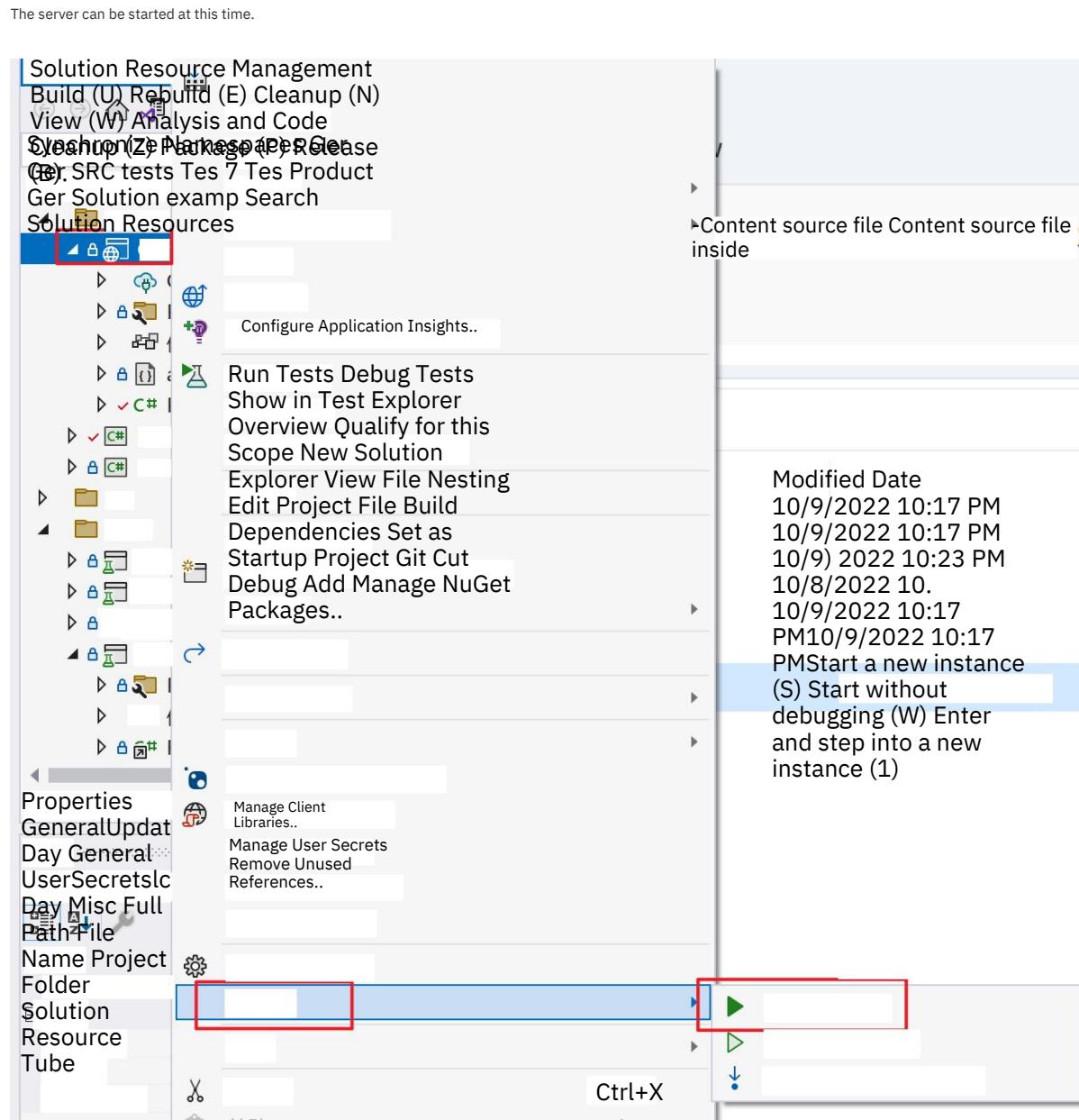


But here I want to let you feel the effect of the update, and then package the contents of the update package is some txt text files (practical use can skip the following link, just to demonstrate the effect of the update).

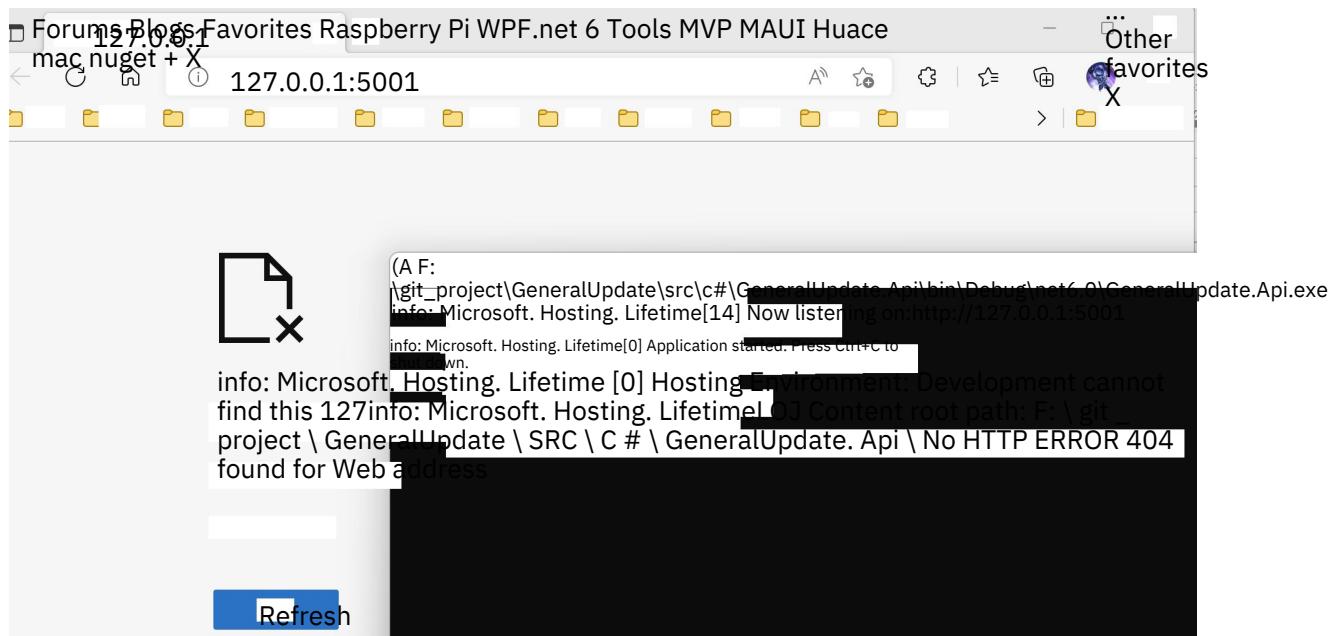
You also need to copy all the file contents here to the Run directory.

A screenshot of a file explorer window. The path is shown as "This computer &gt; F (F:) &gt; temp &gt; source name temp 1.txt". The file is described as a "Source file content 1.txt-Notepad File". The file's size is 0, it was modified on 10/6/2022 at 6:51 PM, and it is a Text Document. The file icon is a blue document with horizontal lines. The window has a standard OS X-style interface with a toolbar and status bar.

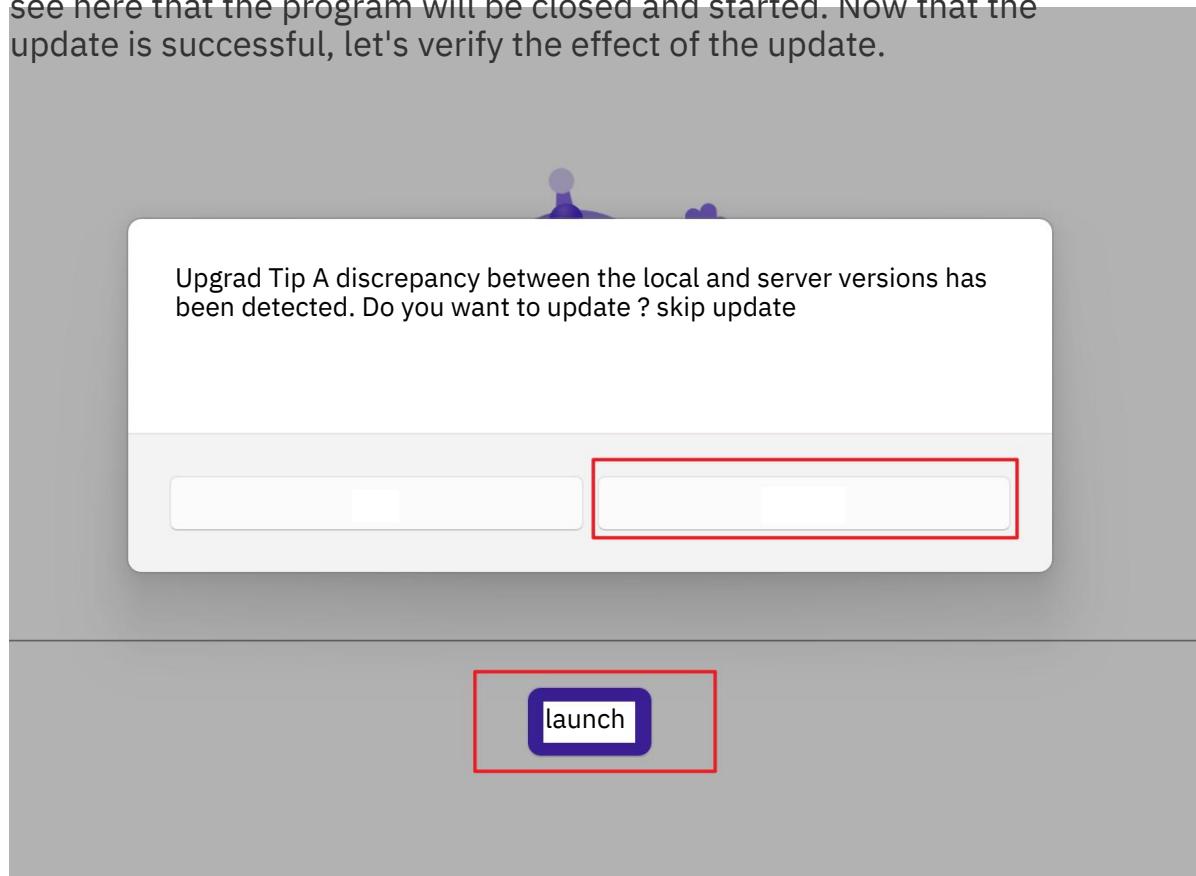
> This computer > E (E:) > Run > Name		9 Search size in Run
sr-CyrL-RS sr-Latn- RS SV-SE temp th - TH tr-TR uk-UA vi- VN zh-CN zh-TW 1.txt 123. BMP AboutAssets. Txt	Modified Date	Type
	10/9/2022	Folders
	10:17 PM	Folders
	10/9/2022	Folder
	10:17 PM	Folders
	10/9/2022	Folders
	10:23 PM	Folder
	10/8/2022	Folders
	10/9/2022	folder
	10:17 PM	Folder Text
	10/9/2022	Document
	10:17 PM	BMP File
	10/9/2022	Text
	10:17 PM	Document
	10/9/2022	
	10:17 PM	
	9/23/2022	
	11:16 PM	
	9/25/2022	
	6:14 PM	1KB
	9/14/2022	0KB
	11:02 PM	1KB

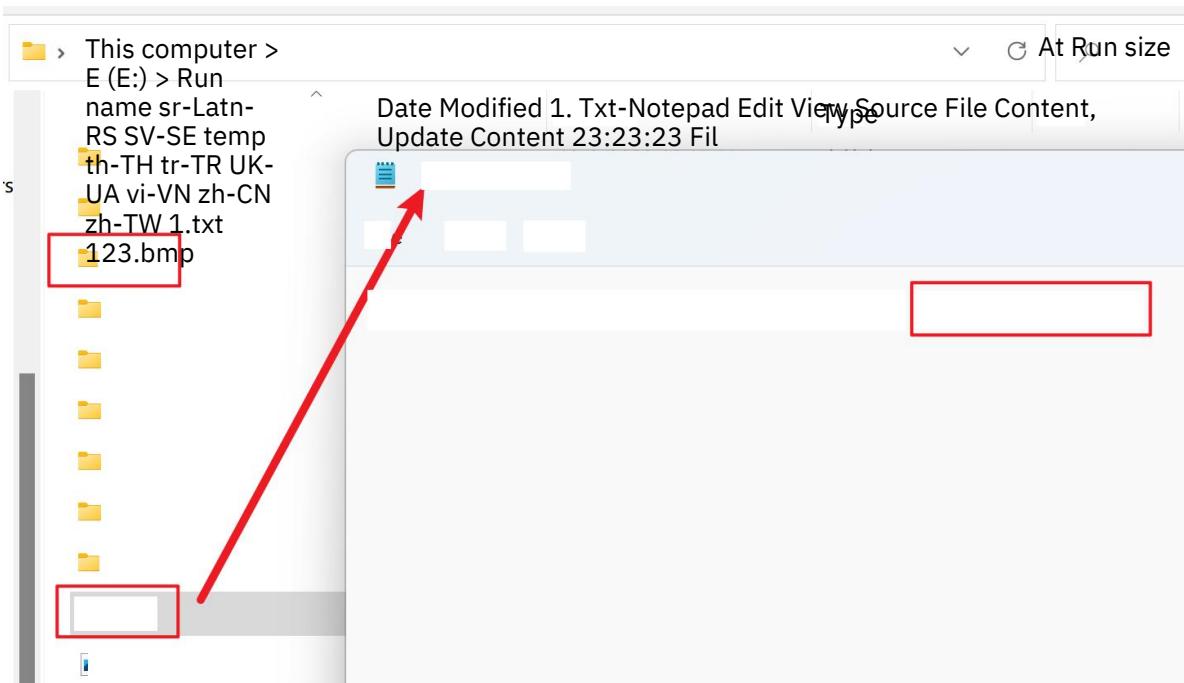


This is the case after successful startup.

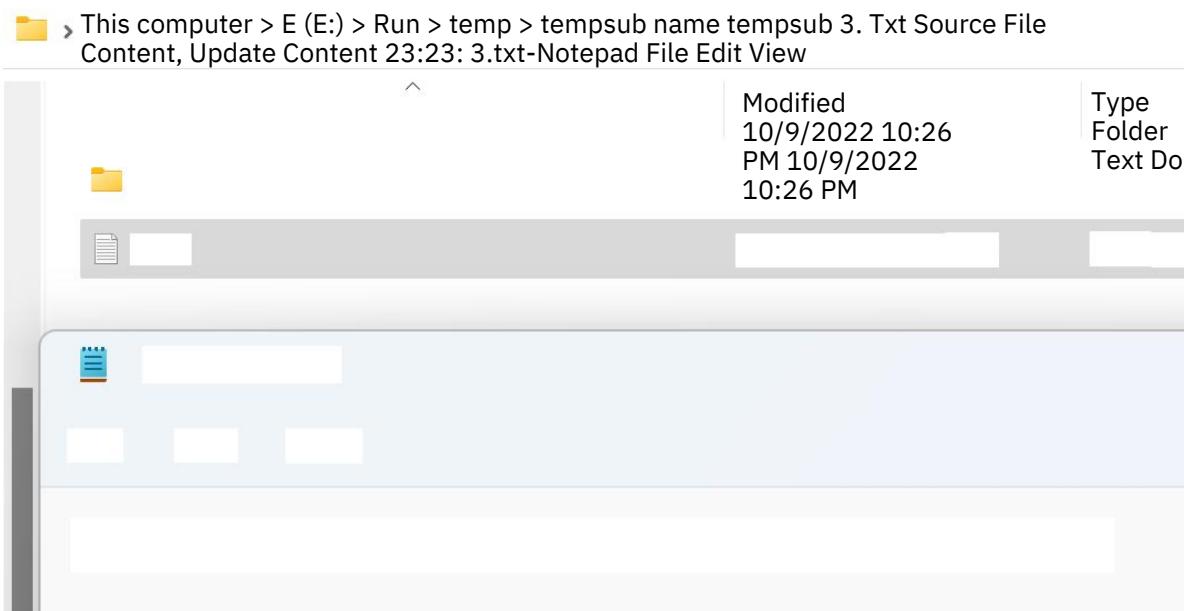


Then click launch on the GeneralUpdate. Client. Exe. I believe you can see here that the program will be closed and started. Now that the update is successful, let's verify the effect of the update.





Then see if the contents of the multiple nested levels of the temp folder are updated to.

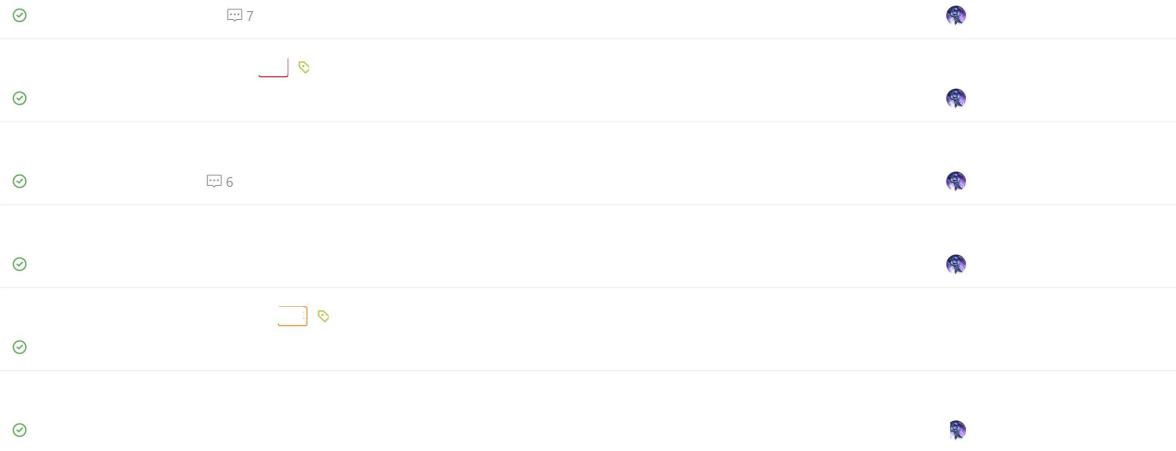


As you can see from the content in the file, there are some new additions. Congratulations on learning the basic use.

## 12.Q&A 1. Comprehensive Q & A reference

Address: [Issues · Juster. Zhu/GeneralUpdate-Gitee. Com](#)

All Opened 21 General update Projected 5 Completed 55 />  
Most 2 seen Multi-Platform All-Search Problem Encountered Bug Responsible Issues with xagan  
Most to Zhu # 1590 Juster. Zhu Chinese name shows garbled after file decompression All-  
Platform bug Critical Owner: 6 months ago # 1502QQ Juster. Zhu removed GeneralUpdate.  
Common component Critical feature 7 months ago Juster. Zhu Owner: # I4YNP5 Juster. Zhu  
Reconstructed Document Function System All-Platform feature iterate Principal: 4 months ago  
# 159Q5W 8 Juster. Zhu Juster. Zhu 20220321 New Feature Test Primary 7 months ago #  
14YVMY Juster. Zhu Leader: Juster. Zhu Update local configuration file or SQLite database  
structure feature Primary Juster. Zhu Leader: 8 months ago voontl # Juster. Zhu refactoring  
compression component feature Leader: Juster. Zhu 8 months ago # | 4TGJJ Juster. Zhu  
refactoring multitasking download feature Main P 00 Primary 1 1 5 Starred Watching 8 F o R K 402



## 2. Common Q & A

(1) Is mutual upgrade supported between the main program and the upgrade program?  
A: Yes.

(2) Do developers need to write code to close the process or other code? A: No, the component already takes into account the entire update process. So there is no need for developers to write any additional auxiliary code besides the component code.

(3) Does the update program need to be placed in the same directory as the main program? A: Yes, we do. But be sure to keep the upgrade program from referencing any code in the main program. Otherwise, the update fails.

(4) Will the patch file of the update package be deleted after the update is completed?  
Answer: Yes, after the component update is completed, it will ensure that the file list is clean, and there will be no redundant file pollution and disk space occupation.

(5) Can it be used on the server side? It is the upgrade between services.

Answer: Theoretically supported, the author has not actually used it in this way. According to the feedback, some partners have already done so. This sharing is for C/architecture scenarios.

(6) How to get the MD5 code of the update package? Answer: Use the AutoUpdate.MD5 project in the project source code.

How to make an update package? Answer: Use the GeneralUpdate.PacketTool tool to generate it. The packaged installer can be seen in the release of the source repository.

(8) How to learn about other components? A: You can communicate with me through the official website, or related Q groups, as well as my gitee or github issues.

(9) When the download package is decompressed to Program Files (x86) under the C disk, how to deal with the operation without permission? Answer:  
<https://gitee.com/Juster-zhu/GeneralUpdate/issues/I4ZKQ4>

---

(10) When the file is updated, the download speed is 0B/S.

Answer: <https://gitee.com/Juster-zhu/GeneralUpdate/issues/I3POMG>  
Keywords: C/S, WPF, MAUI, Winfrom, Avalonia, Console App, UWP, WinUI, Linux, Windows, MacOS, Auto Update, Auto Upgrade, Update, Push.