راهنمای برنامه نویسی برای شناسایی قفل

فهرست :

برای برقراری ارتباط بین برنامه ی شما و قفل سخت افزاری نیاز دارید که فایل ActiveX منشورسیمین را روی کامپیوتری که میخواهید با آن برنامه نویسی کنید رجیستر کنید.

چنانچه مایل نبودید روی سیستم کاربر نهایی تان هم این فایل ActiveX را رجیستر کنید باید از Manifest استفاده کنید که شیوه ی استفاده از ان در ادامه توضیح داده خواهد شد.

❖ رجیستر کردن ACTIVEX

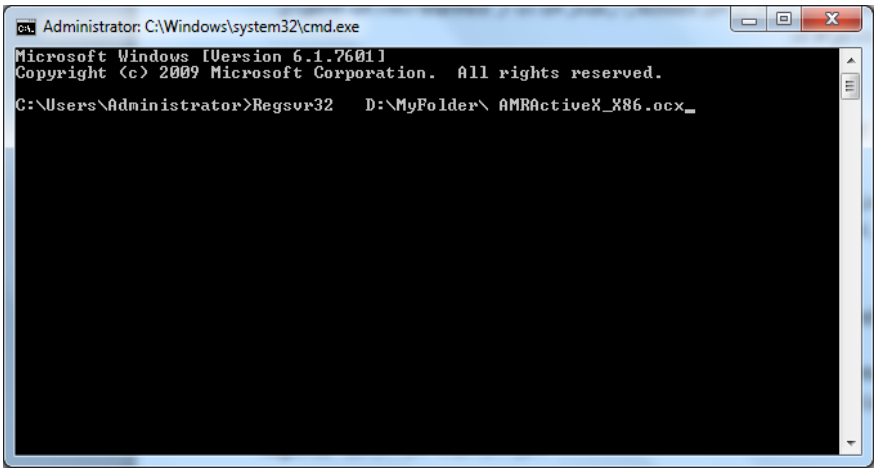
برای رجیستر کردن فایل ActiveX پیشنهاد میشود کاربر Administrator سیستم شما فعال باشد و در این کاربری ActiveX را رجیستر کنید اما گاهی،صرفا با اجرای Command Prompt به صورت Run As Administrator مشکل حل میشود.

- رجیستر کردن ActiveX در ویندوز ۳۲ بیتی:

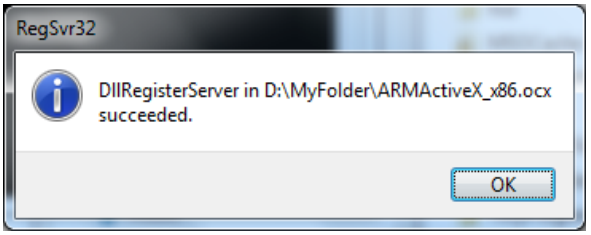
فایل AMRActiveX_X86 را از روی سایت دانلود کرده سپس آن را در مسیر ثابتی روی هارد خود قرار دهید،سپس Command Prompt را اجرا کرده و دستور زیر را در آن تایپ کنید :

مسیر کامل فایل به همراه نام فایل Regsvr32

برای مثال :



با اجرای دستور فوق باید ویندوز به شما پیغام



را نمایش دهد.

***توجه کنید در مسیر فایل ActiveX نباید پوشه ای با نام فارسی قرار گرفته باشد.

- رجیستر کردن ActiveX در ویندوز ۶۴ بیتی:

جهت رجیستر کردن اکتیوایکس روی ویندوز ۶۴ بیتی ، بهتر است فایل اکتیوایکس **۳۲** بیتی را از روی سایت دانلود کنید و سپس آن را در مسیر C:\Windows\SysWOW64 کپی کنید سپس از این مسیر برای رجیستر کردن اقدام کنید.

برای رجیستر کردن به این صورت اقدام کنید : Command Prompt ویندوز را باز کنید (ترجیحا Runas Administrator) در خط دستور تایپ کنید :

مسیر کامل فایل به همراه نام فایل Regsvr32

برای مثال: regsvr32 C:\Windows\SysWOW64\ ARMActiveX(X86 Ver1.1.1).ocx

با اجرای این خط ویندوز باید به شما پیغام Succeeded بدهد!

سپس در پروژه ی خود (چنانچه در محیط ویژوال استودیو کدنویسی می کنید platformرا روی x86 ست کنید .)در نهایت فایل exe تولیدی شما هم روی ویندوزهای ۶۴ بیتی قابل اجرا خواهد بود هم روی ویندوزهای ۳۲ بیتی.
اما اگر نیاز داشتید که Platformروی anycpu یا روی x64 باشد علاوه بر اکتیوایکس ۳۲ بیتی اکتیوایکس ۶۴ بیتی را هم باید رجیستر کنید. به این صورت که :

فایل AMRActiveX_X64 را از روی سایت دانلود کرده سپس آن را در مسیر ثابتی روی هارد خود قرار دهید،سپس Command Prompt را اجرا کرده و دستور زیر را در آن تایپ کنید :

Regsvr32 مسیر کامل فایل به همراه نام فایل

برای مثال :

Regsvr32 D:\MyFolder\ AMRActiveX_X64.ocx

با اجرای دستور فوق باید ویندوز به شما پیغام succeeded دهد.

بعد از رجیستر کردن ActiveX بستر برنامه نویسی برای شما فراهم میشود.

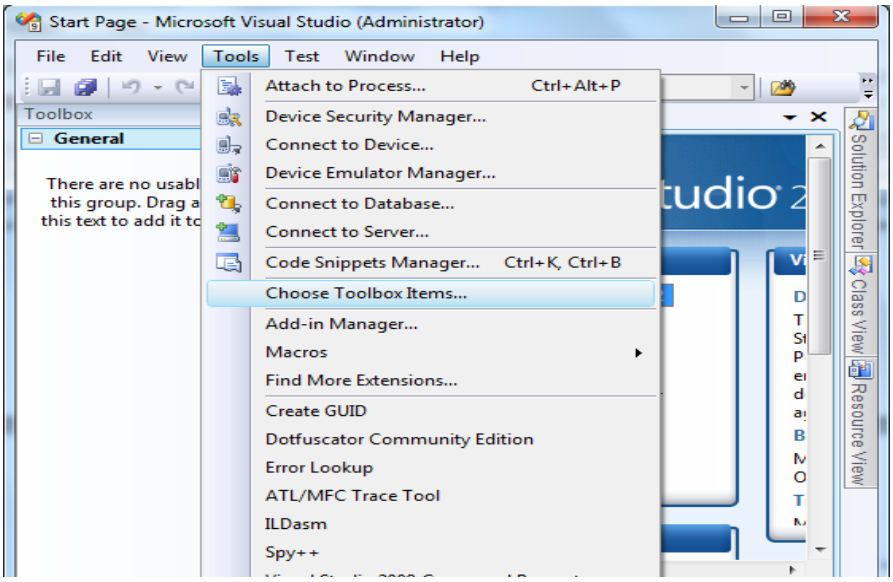
❖ اضافه کردن کنترل ACTIVEX به محیط برنامه نویسی C#

حال با توجه به محیطی که در آن برنامه نویسی میکنید باید کنترلActiveX را به پروژه ی خود اضافه کنید.

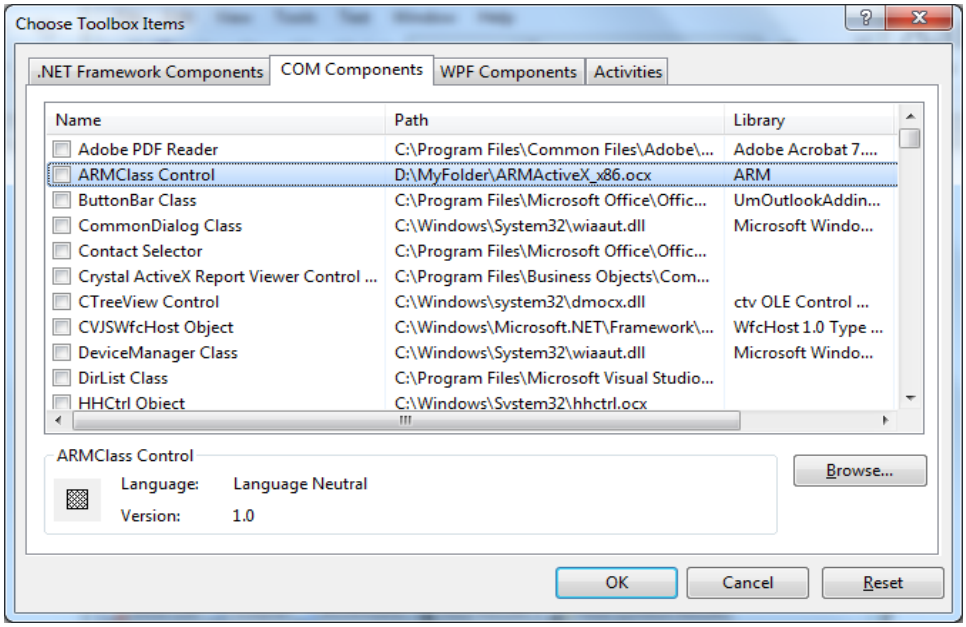
مثلا در C# این کار بدین صورت انجام میشود :

- اضافه کردن کنترل ActiveX روی فرم :

از منوی Tools گزینه ی "Choose Toolbox Items..." را انتخاب کنید به صورت زیر :



از پنجره ی باز شده در Tab ، COM Component گزینه ی ARMClass Control را تیک بزنید و ok کنید به صورت زیر



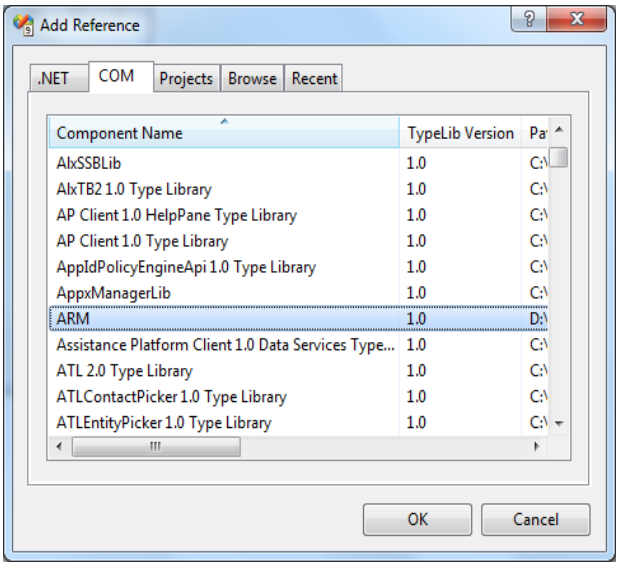
با این کار کنترل ARM به ToolBox شما اضافه میشود و شما میتوانید با اضافه کردن کنترل ،به فرم خود،از توابع از پیش تعریف شده برای شناسایی قفل استفاده کنید که شیوه ی کار با این توابع در زیر توضیح داده شده است.

- اضافه کردن کلاس ActiveX به پروژه و استفاده از Objectآن :

اما چنانچه مایل نبودید کنترل را به فرم خود اضافه کنید و فقط میخواستید از کلاس آن یک Object بسازید و با آن کار کنید به صورت زیر اقدام میکنید :

در محیط Visual Studio در پنجره Solution Explorer کلیک راست کرده و از منوی باز شده گزینه Add Reference را انتخاب نمایید

در پنجره باز شده Tab Control ، Com را انتخاب کرده و از لیست، گزینه ARM را انتخاب کنید



پس از کلیک روی دکمه OK ماژول ARM به پروژه اضافه میشود. پس از این مرحله می توانید با تعریف یک Object از این ماژول به توابع و متغییر های این ماژول دسترسی پیدا کنید.

- ## اضافه کردن کلاس ActiveX به یک پروژه ی DLL و استفاده از DLL در پروژه ی اصلی :

چنانچه مایل بودید چک قفل را در یک dll انجام دهید باید به شیوه ی زیر عمل کنید :

به پروژه ی dll خود یک Form اضافه کنید به فرم اضافه شده یک بار کنترل ARM را از Toolbox اضافه کنید سپس آنرا پاک کنید،با این کار AXARM به Refrence های پروژه ی شما اضافه میشود . حال در کد های خود،در آن بخشی که میخواهید Object ای از کلاس ARM تعریف کنید به شیوه ی زیر عمل کنید :

```
namespace ClassLibrary1
{
    public class Class1
    {
        public AxARM.AxARMClass arm;
        public  Class1()
        {
            Form1 frm = new Form1();
            arm = new AxARM.AxARMClass();
            arm.CreateControl();
            arm.Parent = frm;
        }
        public void CheckLock()
        {
            int ErrorCode = 0;
            arm.FindFirstARM("10D444B57846080B66FD48398B2DBD3829E", "515F9BA9E5
            ErrorCode = arm.GetARMErrorCode();
        }
    }
}
```

ابتدا object را به صورت Global از کلاس AxARMClass تعریف میکنیم.

در سازنده ی کلاس آن را New میکنیم سپس تابع CreateControl را صدا زده بعد برای آن Parent ست میکنیم که این Parent همان فرم خالی ما است.(Object ساخته شده از کلاس AxARMClass حتما باید Parent داشته باشد.)

سپس در تابع شناسایی قفل از توابع آن برای شناسایی قفل استفاده میکنیم.

❖ اضافه کردن کنترل ACTIVEX به محیط برنامه نویسی DELPHI

۱. در دلفی یک Package ایجاد کنید
۲. در Package ایجاد شده از منوی Component گزینه ی Import Component را انتخاب کنید
۳. در پنجره ی باز شده تیک Import ActiveX Control را بزنید و Next کنید.
۴. در کادر search عبارت ARM را تایپ کنید تا ActiveX ای که رجیستر کردید را برای شما پیدا کند.
۵. Next کنید.
۶. در این پنجره Package ای که ایجاد کردید را انتخاب کنید و دکمه ی finish را بزنید.با این کار ActiveX به این Packageاضافه شده است،حال Packageرا Compile کنید و سپس Instal کنید،با این کار کنترل ActiveX به لیست کنترل های شما اضافه خواهد شد و میتوانید از ان استفاده کنید.

❖معرفی توابع ACTIVEX:

Authenticate :

- پارامترها :

۱. uint Random,

این پارامتر باید یک عدد Random کوچکتر از ۲۱۴۷۴۸۳۶۴۷باشد(Unsigned int) که توسط خود برنامه نویس تولید میشود.

قابل ذکر است،چنانچه این عدد رندوم به صورت تکراری به قفل ارسال شود قفل بلاک میشود و دیگر جواب صحیح نمیدهد تا زمانی که قفل را از سیستم جدا کنید و مجدد وصل کنید.

- مقدار بازگشتی :

مقدار بازگشتی این تابع از نوع Object است که این Object را به همراه عددRandom ای که در مرحله ی قبل تولید کردیم به تابع CheckAuthenticate پاس میدهیم و مقدار بازگشتی آن را بررسی میکنیم که این مقدار یک عدد است ،

چنانچه این عدد "۱" بود این بدین معنی است که این مرحله به درستی صورت گرفته است و باید وارد مرحله ی بعدی یعنی FindFirstARM شدامااگر این عدد بغیر از "۱"بود :

"۱۱۳" : به معنی یک خطای ناشناخته است!

"۱۰۱": به معنی پیدا نشدن قفل است (این خطا در این مرحله یعنی یا قفل روی سیستم موجود نیست یا مشکلی برای سخت افزار قفل به وجود آمده است و یا عدد رندومی که به عنوان پاراامتر به تابع Authenticate ارسال کرده ایم تکراری بوده!و ای روی قفل پسوردی ست نشده است!)

"۱۰۶":این خطا در حالت شبکه رخ خواهد داد و به معنی عدم برقراری ارتباط بین اکتیوایکس و سرویس است!

FindFirstARM :

چنانچه مقدار بازگشتی تابعCheckAuthenticate،"۱" بود FindFirstARMرا صدا میزنیم.

- پارامترها :

۱. Object UserKey

۲. String strKey1

۳. String strKey2

برای تولید پارامتر اول از تابع CreateUserKey کمک میگیریم،بدین ترتیب که رشته ای که نرم افزار Manager تحت عنوان ReadKeyو یا Read/Write Key برای ما تولید کرده را به همراه کلید AES ای که در هنگام نوشتن پسورد، روی قفل نوشته ایم را به این تابع پاس میدهیم و Object تولیدی را به عنوان پارامتر اول به تابع FindFirstARM میدهیم.

*تذکر : چنانچه زمان نوشتن پسورد روی قفل کلیدAES را، ست نکرده باشیم در این مرحله به عنوان کلید باید یک رشته ی ۱۶تایی از Space را به تابع ارسال کنیم.

دو پارامتر دیگر توسط نرم افزار Manager تولید میشوند که در فایل Help مربوط به Manager توضیح داده شده اند.

GetARMErrorCode :

چنانچه تمامی مراحل فوق به درستی پیش رفتند میتوان این تابع را صدا زد.

- پارامترها :

  یک آرایه ی ۱۶ تایی از نوع بایت که حاوی مقادیر Random است دریافت میکند که این مقادیر را برنامه نویس خودش تولید کرده و به تابع ارسال میکند.

- مقدار بازگشتی :

  مقدار بازگشتی این تابع از نوع Object است که این مقدار را به همراه کلید AES و رندوم هایی که به GetARMErrorCodeفرستاده بودیم به تابع GenerateErrorCode ارسال میکنیم و کد خطا را دریافت میکنیم. که در ادامه کد خطاهای احتمالی شرح داده خواهند شد.

GetARMData :

- پارامتر ها :

  ۱. WhichData این پارامتر مشخص میکند که شما میخواهید کدام یکی از دیتا های قفل را بخوانید که این انتخاب به کمک Enum "ARM.IwhichData" که در ActiveX تعریف شده است ، انجام میشود.

  ۲. Rprogram : این پارامتر یک عدد Random از نوع Unsigned int که کوچکتر از ۲۱۴۷۴۸۳۶۴۷ است.

- مقدار بازگشتی این تابع از نوع object است ، این مقدار را به همراه Rprogram به تابع DecodeData ارسال میشود در این تابع عملیات رمز گشایی روی دیتای دریافتی از قفل انجام میشود و دیتای نهایی به صورت string برگردانده میشود.

SetARMData :

ابتدا دیتایی از قفل که میخواهیم در آن رایت کنیم را مشخص کرده به همراه رشته ای که میخواهیم به عنوان دیتا روی قفل رایت کنیم و کلیدAES را به تابع CodeData ارسال میکنیم ،مقدار بازگشتی این تابع از نوع object است.

- پارامتر ها :

۱. WhichData : این پارامتر مشخص میکند که شما میخواهید کدام یک از دیتاهای قفل را رایت کنید و این انتخاب به کمک Enum "ARM.IwhichData" که در ActiveX تعریف شده است ، انجام میشود.

۲. DataToWrite : Object ای که از تابع CodeData گرفتیم را به عنوان پارامتر دوم در نظر میگیریم.

- این تابع مقدار بازگشتی ندارد.

GetARMEqu_Result :

در صورتی از این تابع در برنامه ی خود استفاده کنید که برای قفل معادله ست کرده باشید.

- پارامترها :

ابتدا ۴ متغیر از نوع Integer ، که این ۴ متغیر به عنوان پارامترهای معادله در نظر گرفته میشوند به همراه کلید AESای که در ابتدا روی قفل ذخیره کرده ایم به تابع `EnCodeDataEquation` ارسال میکنیم و مقداری که این تابع برمیگرداند را به عنوان پارامتر به تابع GetARMEqu_Result ارسال میشود و قفل جواب معادله ای که توسط برنامه نویس ست شده را بدست میاورد و برای برنامه ارسال میکند و برنامه نویس هم در برنامه ی خود با این ۴ پارامتر جواب معادله را بدست میاورد و با جوابی که از سمت قفل آمده مقایسه میکند

- مقدار بازگشتی : این تابع مقدار Long برمیگرداند.

GetARMExtraData :

- این تابع پارامتر ندارد.
- مقدار بازگشتی این تابع از نوع object است که این Object را به صورت آرایه ای از بایت در نظر میگیریم.

SetARMExtraData :

- به عنوان پارامتر آرایه ای از بایت را که میخواهیم به عنوان ExtraData روی قفل رایت شود رابه تابع ارسال میکنیم
- این تابع مقدار بازگشتی ندارد.

GetARMQuery :

- پارامتر :

این تابع یک پارامتر از نوع object میگیرد که برای بدست آوردن این پارامتر یکی از عناصر آرایه ی ArrRequest (مثلا عنصر iام) را انتخاب کرده و آن را به عنوان string به تابع `EncodeQueryRequest` ارسال میکنیم و مقدار بازگشتی این تابع را که یک Object است را به تابع GetARMQuery میدهیم. که این انتخاب هر بار به صورت Random اتفاق می افتد.(آرایه ی ArrRequest توسط نرم افزار Manager تولید شده است و شما باید این آرایه را در کد های برنامه ی خود تعریف کنید.)

- مقدار بازگشتی این تابع از نوع Object است و آن را به همراه کلید AES به تابع `DecodeQueryResponse` ارسال میکنیم و مقدار بازگشتی آن که از نوع string است باید برابر با عنصر متناظر از آرایه ی ArrResponse باشد. .(آرایه ی ArrResponse هم توسط نرم افزار Manager تولید شده است و شما باید این آرایه را در کد های برنامه ی خود تعریف کنید.)

برای مثال ، شما به عنوان پارامتر عنصر i ام را به تابع ارسال کرده اید حالا برای جواب تابع باید منتظر عنصر iام از تابع ArrResponse باشید.

FindNetARM :

- پارامتر :

۱.ipServer : در این پارامتر همانطور مکه از نام ان پیداست Server ip را وارد کنید.

۲.UserKey : در این پارامتر که از نوع string است ، UserKey قفل را وارد کنید.

- این تابع مقدار بازگشتی ندارد.

GetARMVersion :

- این تابع پارامتری ندارد.
- مقدار بازگشتی این تابع از نوع string است و ورژن ActiveX ای که دارد به شما جواب برمیگرداند را نمایش میدهد.

شرح کدهای خطا

خطای شماره ۱۰۰
این شماره خطا بدین معنی است که همه چیز صحیح است(هم قفل به سیستم متصل است هم کلید ارسالی صحیح است و هم کلید AES ای که استفاده شده است.)

خطای شماره ۱۰۲
این شماره خطا بدین معنی است که کلیدی که برای شناسایی به قفل ارسال شده کلید صحیحی نیست.

خطای شماره ۱۰۴
این شماره خطا بدین معنی است که یکی از strkey1یا strkey2 صحیح نیست.

خطای شماره ۱۰۵
این شماره خطا بدین معنی است که مشکلی برای قفل پیش آمده و باید به شرکت برگردد.

خطای شماره۱۰۶
این خطا در حالت شبکه رخ میدهد و بدین معنی است که ارتباط سرور با کلاینت به درستی برقرار نمیشود! در این حالت ابتدا از صحت ارتباط بین کلاینت و سرور مطمئن شوید سپس مطمئن شوید که روی سرورFireWall خاموش باشد و یا پورت ۹۰۵۱ را باز بگذارید(سرویسARM با پورت ۹۰۵۱ کار میکند)همچنین اگر آنتی ویروس دارید مطمئن شوید که آنتی ویروس این پورت را نمیبندد! سپس وضعیت سرویس نصب شده روی سرور را بررسی کنید اگر Stopشده Start کنید و اگر Disable شده از TaskManager پروسس ArmService را EndTask کنید سپس مجدد سرویس را start کنید. در صورت ادامه داشتن مشکل،با شرکت تماس بگیرید.

خطای شماره۱۰۷
این خطا در حالت شبکه رخ میدهد و بدین معنی است که تعداد کاربران مجاز شبکه(NTUser) روی قفل ست نشده است!این آیتم در نرم افزار ARMManager در بخشVariablesData قابل دسترس است.

خطای شماره۱۱۱
این شماره خطا بدین معنی است که از قفل به صورت غیر استاندارد استفاده میشود!

خطای شماره۱۱۳
در هنگام رخ دادن این خطا با شرکت تماس بگیرید!

خطای شماره۲۰۰
این شماره خطا بدین معنی است که یا کلید AES به کار رفته اشتباه است و یا UserKey !