

عنوان: ویدیوهای آموزشی QT
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۴/۲۵ ۰۹:۱۶:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: ++C

در کمال ناباوری، یک مجموعه‌ی آموزشی 88 قسمتی، از فریم ورکی به نام [QT](#) را در یوتیوب پیدا کردم [[+](#)] که نتیجه‌اش دریافت و آپلود مجدد آن‌ها جهت سهولت دریافت بود. حجم کل آن‌ها 2.14 گیگابایت است که در 13 قسمت جهت علاقمندان آپلود شده است (مجوز ویدیوهای یوتیوب، Creative Commons است و امکان توزیع مجدد آن‌ها وجود دارد: [[+](#)] و [[+](#)]). برای باز کردن مجموعه فایل‌های دریافت شده از برنامه [Zip-7](#) می‌شود استفاده کرد.

[دریافت قسمت‌های 1 تا 88](#)

نظرات خوانندگان

نویسنده: Mehdi

تاریخ: ۲۰:۴۸:۰۴ ۱۳۹۰/۰۴/۲۵

با سلام و تشکر بابت آموزش های خوبی که در سایتتون دارید.
آیا این ویدیو ها برای یه فرد که تازه می خواد Qt یاد بگیره مناسبه و اینکه آیا این ویدیو ها دارای یه نظم سلسه وار در آموزش هست یا به صورت پراکنده هستش.

نویسنده: وحید نصیری

تاریخ: ۲۱:۰۶:۰۴ ۱۳۹۰/۰۴/۲۵

لیست عناوین ویدیوها رو میتونید اینجا مشاهده کنید:
<http://www.voidrealms.com/tutorials.aspx?filter=qt>

نویسنده: Alinn

تاریخ: ۱۳:۳۴:۴۲ ۱۳۹۰/۰۴/۲۶

سلام
عیدتون مبارک!
ممنون برای آپلود مجدد این فایلها. متاسفانه سایتی که توش آپلود کردید فیلتره:)

نویسنده: وحید نصیری

تاریخ: ۱۴:۱۷:۰۵ ۱۳۹۰/۰۴/۲۶

هات فایل فیلتر نیست. به همین جهت این مورد هم ذکر شد.

نویسنده: Reza74m

تاریخ: ۲۰:۰۴:۲۷ ۱۳۹۰/۰۴/۲۷

سلام، اول تشکر
من از طریق دنبال کردن فید شما این 88 تا رو خودم از یوتیوب دانلود کردم ولی حجم فایلها 1.63 گیگا شد به فرمت f1v! یا فرمتها فرق میکنه یا یکی اشتباه میکنه!

نویسنده: وحید نصیری

تاریخ: ۲۱:۰۸:۵۹ ۱۳۹۰/۰۴/۲۷

ویدیوهای یوتیوب با چندین کیفیت مختلف ارائه می‌شن (360، 480، 720 و الی آخر [+](#)). پیش فرض عموماً 360 است. این آپلود فوق بهترین کیفیت ممکن را دارد.

نویسنده: Mostafa

تاریخ: ۰۶:۲۱:۱۲ ۱۳۹۰/۰۵/۰۹

با تشکر
برای باز کردن پسورد میخواد
پسوردش چیه؟

نویسنده: وحید نصیری

تاریخ: ۰۸:۱۵:۳۰ ۱۳۹۰/۰۵/۰۹

پسورد نداره. یا ناقص دریافت کردید یا از برنامه zip-7 برای باز کردن استفاده نکردید. مورد اول بیشتر محتمل است.

نویسنده: Masoud Lotfi
تاریخ: ۱۳:۴۸:۵۸ ۱۳۹۰/۰۶/۰۶

اول تشکر می کنم بابت زحمتی که برای جمع اوری این آموزش کشیدید.
دوم یه سوالی داشتم. من فایل ها رو از هات فایل دانلود کردم ، اما مشکلی که دارم این که شماره فایل ها به عنوان پسوند فایل قرار گرفته و خوب بالطبع نمی تونم با برنامه وینرر بازشون کنم. باید چی کار کنم؟ ممنون میشم کمک کنید.

نویسنده: وحید نصیری
تاریخ: ۱۴:۱۴:۲۹ ۱۳۹۰/۰۶/۰۶

از برنامه رایگان zip-7 استفاده کنید: <http://www.7-zip.org> ، با همین برنامه فشرده شده اند.

نوع شمارشی enum

نوع شمارشی، یک نوع صحیح است و شامل لیستی از ثوابت می‌باشد که توسط برنامه نویس مشخص می‌گردد. انواع شمارشی برای تولید کد خودمستند به کار می‌روند یعنی کدی که به راحتی قابل درک باشد و نیاز به توضیحات اضافه نداشته باشد. زیرا به راحتی توسط نام، نوع کاربرد و محدوده مقادیرشان قابل درک می‌باشند. مقادیر نوع شمارشی منحصر به فرد می‌باشند (unique) و شامل مقادیر تکراری نمی‌باشند در غیر این صورت کامپایلر خطای مربوطه را هشدار میدهد. نحوه تعریف نوع شمارشی:

```
enum typename{enumerator-list}
```

enum کلمه کلیدی ست، typename نام نوع جدید است که برنامه نویس مشخص میکند و enumerator-list مجموعه مقادیری ست که این نوع جدید می‌تواند داشته باشد بعنوان مثال:

```
enum Day{SAT,SUN,MON,TUE,WED,THU,FRI}
```

اکنون Day یک نوع جدید است و متغیرهایی که از این نوع تعریف می‌شوند میتوانند یکی از مقادیر مجموعه فوق را دارا باشند.

```
Day day1,day2;
day1 = SAT;
day2 = SUN;
```

مقادیر SAT و SUN و MON هر چند که به همین شکل بکار می‌روند ولی در رایانه به شکل اعداد صحیح 0, 1, 2, ... ذخیره می‌شوند. به همین دلیل است که به هر یک از مقادیر SAT و SUN و ... یک شمارشگر می‌گویند. وقتی فهرست شمارشگرهای یک نوع تعریف شد به طور خودکار مقادیر 0 و 1 و ... به ترتیب به آنها اختصاص داده میشود. می‌توان مقادیر صحیح دلخواهی به شمارشگرها نسبت داد به طور مثال:

```
enum Day{SAT=1,SUN=2,MON=4,TUE=8,WED=16,THU=32,FRI=64}
```

اگر چند شمارشگر مقدار دهی شده باشند آنگاه شمارشگرهایی که مقدار دهی نشده اند، مقادیر متوالی بعدی را خواهند گرفت.

```
enum Day{SAT=1,SUN,MON,TUE,WED,THU,FRI}
```

دستور بالا مقادیر 1 تا 7 را بترتیب به شمارشگرها اختصاص میدهد. میتوان به شمارشگرها مقادیر یکسانی نسبت داد

```
enum Answer{NO=0,FALSE=0,YES=1,TRUE=1,OK=1}
```

ولی نمی‌توان نامهای یکسانی را در نظر گرفت! تعریف زیر بدلیل استفاده مجدد از شمارشگر YES با خطای کامپایلر مواجه می‌شویم.

```
enum Answer{NO=0,FALSE=0,YES=1,YES=2,OK=1}
```

چند دلیل استفاده از نوع شمارشی عبارت است از:

- 1- enum سبب میشود که شما مقادیر مجاز و قابل انتظار را به متغیرهایتان نسبت دهید.
- 2- enum اجازه میدهد با استفاده از نام به مقدار دستیابی پیدا کنید پس کدهایتان خواناتر میشود.

3- با استفاده از enum تایپ کدهایتان سریع میشود زیرا IntelliSense در مورد انتخاب گزینه مناسب شما را یاری میدهد .

چند تعریف از enum :

```
enum Color{RED, GREEN, BLUE, BLACK, ORANGE}
enum Time{SECOND, MINUTE, HOUR}
enum Date{DAY, MONTH, YEAR}
enum Language{C, DELPHI, JAVA, PERL}
enum Gender{MALE, FEMALE}
```

تا اینجا خلاصه ای از enum و مفهوم آن داشتیم

اما تغییراتی که در C++11 اعمال شده : Type-Safe Enumerations

فرض کنید دو enum تعریف کرده اید و به شکل زیر می باشد

```
enum Suit {Clubs, Diamonds, Hearts, Spades};
enum Jewels {Diamonds, Emeralds, Opals, Rubies, Sapphires};
```

اگر این دستورات را کامپایل کنید با خطا مواجه می شوید چون در هر دو enum شمارشگر Diamonds تعریف شده است . کامپایلر اجازه تعریف جدیدی از یک شمارشگر در enum دیگری نمیدهد هر چند برخی اوقات مانند مثال بالا نیازمند تعریف یک شمارشگر در چند enum بر حسب نیاز میباشیم .

برای تعریف جدیدی که در C++11 داده شده کلمه کلیدی class بعد از کلمه enum مورد استفاده قرار میگیرد . به طور مثال تعریف دو enum پیشین که با خطا مواجه میشد بصورت زیر تعریف میشود و از کامپایلر خطایی دریافت نمیکنیم .

```
enum class Suit {Clubs, Diamonds, Hearts, Spades};
enum class Jewels {Diamonds, Emeralds, Opals, Rubies, Sapphires};
```

همچنین استفاده از enum در گذشته و تبدیل آن به شکل زیر بود :

```
enum Suit {Clubs, Diamonds, Hearts, Spades};
Suit var1 = Clubs;
int var2 = Clubs;
```

یک متغیر از نوع Suit بنام var1 تعریف میکنیم و شمارشگر Clubs را به آن نسبت میدهیم ، خط بعد متغیری از نوع int تعریف نمودیم و مقدار شمارشگر Clubs که 0 می باشد را به آن نسبت دادیم . اما اگر تعریف enum را با قوائد C++11 در نظر بگیریم این نسبت دادن باعث خطای کامپایلر میشود و برای نسبت دادن صحیح باید به شکل زیر عمل نمود .

```
enum class Jewels {Diamonds, Emeralds, Opals, Rubies, Sapphires};
Jewels typeJewel = Jewels::Emeralds;
int suitValue = static_cast<int>(typeJewel);
```

همانطور که مشاهده میکنید ، Type-Safe یودن enum را نسبت به تعریف گذشته آن مشخص می باشد .

یک مثال کلی و جامع تر :

```
// Demonstrating type-safe and non-type-safe enumerations
#include <iostream>
using std::cout;
using std::endl;
// You can define enumerations at global scope
//enum Jewels {Diamonds, Emeralds, Rubies}; // Uncomment this for an error
enum Suit : long {Clubs, Diamonds, Hearts, Spades};
int main()
{
    // Using the old enumeration type...
    Suit suit = Clubs; // You can use enumerator names directly
```

```
Suit another = Suit::Diamonds; // or you can qualify them
// Automatic conversion from enumeration type to integer
cout << "suit value: " << suit << endl;
cout << "Add 10 to another: " << another + 10 << endl;
// Using type-safe enumerations...
enum class Color : char {Red, Orange, Yellow, Green, Blue, Indigo, Violet};
Color skyColor(Color::Blue); // You must qualify enumerator names
// Color grassColor(Green); // Uncomment for an error
// No auto conversion to numeric type
cout << endl;
<< "Sky color value: " << static_cast<long>(skyColor) << endl;
//cout << skyColor + 10L << endl; // Uncomment for an error
cout << "Incremented sky color: "
<< static_cast<long>(skyColor) + 10L // OK with explicit cast
<< endl;
return 0;
}
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۱۰ ۲۳:۳

خودمونیم! بد طراحی شده. از المان یک enum می‌شده/میشه مستقیماً خارج از enum بدون ارجاعی به اون استفاده کرد؟! به این می‌گن بیش از حد دست و دلبازی و منشاء سردرگمی (که در نگارش 11 به اسم type-safety بالاخره رفع و رجوعش کردن).

variable

متغیر :

برنامه هایی که نوشته می‌شوند برای پردازش داده‌ها بکار می‌روند، یعنی اطلاعاتی را از یک ورودی می‌گیرند و آنها را پردازش میکنند و نتایج مورد نظر را به خروجی می‌فرستند . برای پردازش ، لازم است که داده‌ها و نتایج ابتدا در حافظه اصلی ذخیره شوند، برای این کار از متغیر استفاده میکنیم .

متغیر مکانی از حافظه ست که شامل : نام ، نوع ، مقدار و آدرس می‌باشد . وقتی متغیری را تعریف میکنیم ابتدا با توجه به نوع متغیر ، آدرسی از حافظه در نظر گرفته می‌شود، سپس به آن آدرس یک نام تعلق میگیرد. نوع متغیر بیان میکند که در آن آدرس چه نوع داده ای می‌تواند ذخیره شود و چه اعمالی روی آن می‌توان انجام داد، مقدار نیز مشخص میکند که در آن محل از حافظه چه مقداری ذخیره شده است . در ++C قبل از استفاده از متغیر باید آن را اعلان نماییم . نحوه اعلان متغیر به شکل زیر می‌باشد :

```
type name initializer ;
```

عبارت type نوع متغیر را مشخص میکند . نوع متغیر به کامپایلر اطلاع میدهد که این متغیر چه مقداری می‌تواند داشته باشد و چه اعمالی می‌توان روی آن انجام داد . عبارت name نام متغیر را نشان میدهد. عبارت initializer نیز برای مقداردهی اولیه استفاده می‌شود. نوع هایی که در ویژوال استادیو 2012 ساپورت می‌شوند شامل جدول زیر می‌باشند .

TYPE	SIZE IN BYTES	RANGE OF VALUES
bool	1	true or false
char	1	By default, the same as type signed char: -128 to 127. Optionally, you can make char the same range as type unsigned char.
signed char	1	-128 to 127
unsigned char	1	0 to 255
wchar_t	2	0 to 65,535
short	2	-32,768 to 32,767
unsigned short	2	0 to 65,535
int	4	-2,147,483,648 to 2,147,483,647
unsigned int	4	0 to 4,294,967,295
long	4	-2,147,483,648 to 2,147,483,647
unsigned long	4	0 to 4,294,967,295
long long	8	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long long	8	0 to 18,446,744,073,709,551,615
float	4	$\pm 3.4 \times 10^{\pm 38}$ with approximately 7-digits accuracy
double	8	$\pm 1.7 \times 10^{\pm 308}$ with approximately 15-digits accuracy
long double	8	$\pm 1.7 \times 10^{\pm 308}$ with approximately 15-digits accuracy

چند تعریف از متغیر به شکل زیر :

```
int sum(0); // یا int sum=0;
char ch(65); // ch is A
float pi(3.14); // یا float pi = 3.14;
```

همانطور که مشهود می‌باشد طبق تعریف متغیر ، نوع و نام و مقدار اولیه (اختیاری) ، مشخص گردیده است . تا قبل از C++11 تعریف نوع متغیر الزامی بود در غیر این صورت با خطای کامپایلر مواجه می‌شدیم .

تغییرات اعمال شده در C++11 : معرفی کلمه کلیدی **auto**

در C++11 کلمه کلیدی **auto** معرفی و اضافه گردید ، با استفاده از **auto** ، کامپایلر این توانایی را دارد که نوع متغیر را از روی مقدار دهی اولیه آن تشخیص دهد و نیازی به مشخص نمودن نوع متغیر نداریم .

```
int x = 3;
auto y = x;
```

در تعریف فوق ابتدا نوع متغیر x را int در نظر گرفتیم و مقدار 3 را به آن نسبت دادیم. در تعریف دوم نوع متغیر را مشخص نکردیم و کامپایلر با توجه به مقدار اولیه ای که به متغیر y نسبت دادیم، نوع آن را مشخص میکند. چون مقدار اولیه آن x می باشد و x از نوع int می باشد پس نوع متغیر y نیز از نوع int در نظر گرفته می شود.

دلایلی برای استفاده از auto:

Robustness: (خوشفکری) به طور فرض زمانی که مقدار برگشتی یک تابع را در یک متغیر ذخیره میکنید با تغییر نوع برگشتی تابع نیازی به تغییر کد (برای نوع متغیر ذخیره کننده مقدار برگشتی تابع) ندارید.

```
int sample()
{
    int result(0);
    // To Do ...
    return result;
}

int main()
{
    auto result = sample();
    // To Do ...
    return 0;
}
```

و زمانی که نوع برگشتی تابع بنا به نیاز تغییر کرد

```
float sample()
{
    float result(0.0);
    // To Do ...
    return result;
}

int main()
{
    auto result = sample();
    // To Do ...
    return 0;
}
```

همانطور که مشاهده میکنید با اینکه کد تابع و نوع برگشتی آن تغییر یافت ولی بدنه main تابع هیچ تغییری داده نشد.

Usability: (قابلیت استفاده) نیازی نیست نگران نوشتن درست و تایپ صحیح نام نوع برای متغیر باشیم

```
float f(0.0); // خطای نام نوع گرفته می شود
auto f(0.0);  // نیازی به وارد نمودن نوع تایپ نیستیم
```

Efficiency: برنامه نویسی ما کارآمدتر خواهد بود

مهمترین استفاده از auto سادگی آن است.

استفاده از auto بخصوص زمانی که از STL و templates استفاده میکنیم، بسیار کارآمد می باشد و بسیاری از کد را کم میکند و باعث خوانایی بهتر کد می شود.

فرض کنید که نیاز به یک iterator جهت نمایش تمام اطلاعات کانتینری از نوع map داریم باید از کد زیر استفاده نماییم (کانتینر را map در نظر گرفتیم)

```
map<string, string> address_book;
address_book[ "Alex" ] = "example@yahoo.com";
```

برای تعریف یک iterator به شکل زیر عمل میکنیم .

```
map<string, string>::iterator itr = address_book.begin();
```

با استفاده از auto کد فوق را میتوان به شکل زیر نوشت

```
auto itr = address_book.begin();
```

(کانتینرها: containers): کانتینرها اشیایی هستند که اشیا دیگر را نگهداری میکنند و دارای انواع مختلفی می‌باشند به عنوان مثال (vector, map ... ,
(تکرار کننده‌ها: iterators): تکرار کننده‌ها اشیایی هستند که اغلب آنها اشاره گرند و با استفاده از آنها میتوان محتویات کانتینرها را همانند آرایه پیمایش کرد)

فرض کنید که از یک برنامه‌ی native ویندوز برای تهیه تصاویر سایت‌ها در یک برنامه‌ی وب استفاده می‌کنید و صبح که به سایت سر زده‌اید پیام در دسترس نبودن سایت قابل مشاهده است. مشکل از کجا است؟!

یک مثال ساده

```
using System;

namespace AccessViolationExceptionSample
{
    class Program
    {
        private static unsafe void AccessViolation()
        {
            byte b = *(byte*)(8762765876);
        }

        static void Main(string[] args)
        {
            try
            {
                AccessViolation();
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex);
            }

            Console.WriteLine("Press a key...");
            Console.ReadKey();
        }
    }
}
```

برنامه‌ی کنسول فوق را پس از فعال سازی Allow unsafe code در قسمت تنظیمات پروژه، کامپایل کرده و سپس آن را خارج از VS.NET اجرا کنید. احتمالاً انتظار دارید که قسمت catch این کد حداقل اجرا شود و سپس سطر «کلیدی را فشار دهید» ظاهر گردد. اما ... خیر! کل پروسه کرش کرده و هیچ پیام خطایی را دریافت نخواهید کرد. اگر به لاگ‌های ویندوز مراجعه کنید پیام زیر قابل مشاهده است:

```
System.AccessViolationException. Attempted to read or write protected memory.
This is often an indication that other memory is corrupt.
```

و این نوع مسایل هنگام کار با کتابخانه‌های C و C++ زیاد ممکن است رخ دهند. نمونه‌ی آن استفاده از WebControl دات نت است یا هر برنامه‌ی native دیگری. در این حالت اگر برنامه‌ی شما یک برنامه‌ی وب باشد، عملاً سایت از کار افتاده است. به عبارتی پروسه‌ی ویندوزی آن کرش کرده و بلافاصله از طرف ویندوز خاتمه یافته است.

چرا قسمت catch اجرا نشد؟

از دات نت 4 به بعد، زمانیکه دسترسی غیرمجازی به حافظه صورت گیرد، برای مثال دسترسی به یک pointer آزاد شده، استثنای حاصل، توسط برنامه catch نشده و اجازه داده می‌شود تا برنامه کلاً کرش کند. به این نوع استثناءها Corrupted State Exceptions یا CSE گفته می‌شود. اگر نیاز به مدیریت آن‌ها توسط برنامه باشد، باید به یکی از دو طریق زیر عمل کرد: الف) از ویژگی [HandleProcessCorruptedStateExceptions](#) بر روی متد فراخوان کتابخانه‌ی native باید استفاده شود. برای مثال در کدهای فوق خواهیم داشت:

```
[HandleProcessCorruptedStateExceptions]
static void Main(string[] args)
{
```

ب) و یا فایل کانفیگ برنامه را ویرایش کرده و [چند سطر ذیل](#) را به آن اضافه کنید:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <runtime>
    <legacyCorruptedStateExceptionsPolicy enabled="true" />
  </runtime>
</configuration>
```

در این حالت مدیریت اینگونه خطاها در کل برنامه همانند برنامه‌های تا دات نت 3.5 خواهد شد.

نظرات خوانندگان

نویسنده: سوین
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۹:۴۶

با سلام

من در فایل کانفیگ یه WPF App ، تغییرات گفته شده را قرار دادم و خطای زیر رو در vs داد
.The type initializer for 'System.Windows.Application' threw an exception

و بیرون از vs اصلا اجرا نشد خیلی نیاز دارم به این مورد ، چون یه پروژه دارم که درست اجرا میشه اما بعضی مواقع برنامه کرش میکنه و نمی‌تونم catch کنم . با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۹:۵۹

- مطلب فوق بیشتر مرتبط است به استثناهای کتابخانه‌های native استفاده شده در برنامه‌های دات نت. برای سایر موارد باید در فایل App.xaml.cs [موارد ذیل را](#) بررسی کنید:

```
public partial class App
{
    public App()
    {
        this.DispatcherUnhandledException += appDispatcherUnhandledException;
        AppDomain.CurrentDomain.UnhandledException += CurrentDomain_UnhandledException;
    }
}
```

+ نمونه تنظیم زیر در فایل app.config یک برنامه WPF کار می‌کند (آزمایش شد):

```
<?xml version="1.0"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
  </startup>
  <runtime>
    <legacyCorruptedStateExceptionsPolicy enabled="true" />
  </runtime>
</configuration>
```

نویسنده: سوین
تاریخ: ۱۳۹۲/۱۲/۰۲ ۹:۳۸

با سلام؛ من یه پروژه با WPF نوشتم اما یه ایراد داره و اونم اینکه مثلا فرم 1 رو 20 بار اجرا می‌کنی خطا نمی‌ده اما بار 21 ام برنامه کرش میکنه و اصلا نمیشه catch کرد. متن خطا در Log ویندوز اینه

```
Error 01 :
Application: MyWPFApp.exe
Framework Version: v4.0.30319
Description: The process was terminated due to an unhandled exception.
Exception Info: exception code c0000005, exception address 77D52239

Error 02 :
Faulting application name: MyWPFApp.exe, version: 1.0.0.0, time stamp: 0x52d550ac
Faulting module name: ntdll.dll, version: 6.1.7601.17514, time stamp: 0x4ce7b96e
Exception code: 0xc0000005
Fault offset: 0x00032239
Faulting process id: 0xa28
Faulting application start time: 0x01cf113ae6813d88
Faulting application path: R:\Source\MyWPFApp\bin\Debug\MyWPFApp.exe
Faulting module path: C:\Windows\SYSTEM32\ntdll.dll
Report Id: 460eda62-7d33-11e3-a572-ac220bc99cf8

Information :
```

Fault bucket , type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: MyWPFAApp.exe
P2: 1.0.0.0
P3: 52d550ac
P4: ntdll.dll
P5: 6.1.7601.17514
P6: 4ce7b96e
P7: c0000005
P8: 00032239
P9:
P10:

Attached files:
C:\Users\Administrator\AppData\Local\Temp\WERE9B3.tmp.WERInternalMetadata.xml
C:\Users\Administrator\AppData\Local\Temp\WER16AC.tmp.appcompat.txt
C:\Users\Administrator\AppData\Local\Temp\WER18A1.tmp.hdump
C:\Users\Administrator\AppData\Local\Temp\WER3BFA.tmp.mdmp

These files may be available here:
C:\Users\Administrator\AppData\Local\Microsoft\Windows\WER\ReportQueue\AppCrash_MyWPFAApp.exe_125fc667a69fcc31c463a5e1b4032657c4ce830_cab_0ac03d3e

Analysis symbol:
Rechecking for solution: 0
Report Id: 460eda62-7d33-11e3-a572-ac220bc99cf8

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۱:۳۷

از چه کامپوننتی استفاده کردی ؟ بهتره اون فایل‌های کرش دامپ dmp رو براشون ارسال کنی.

نویسنده: سوین
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۹:۲۵

با سلام
از کامپوننت شرکت‌های ثالث استفاده نکردم . آیا راه حل کلی برای پیدا کردن چنین خطاهایی وجود نداره ، اینترنت رو هم سرچ کردم اما کمک زیادی نکرد که بشه فهمید مشکل از چیه و قبل ارسال این پست 2 ساعت تمام آزمایش کردم خطا نداد اما بعضی مواقع این اتفاق می‌افته .
در ضمن این برنامه WPF App که برای اوتوماسیون اداری نوشته شده و از EF 6.2 ، قفل سخت افزاری (که بدون قفل هم این ایراد رو می‌ده)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۳ ۰:۴۶

مثال WPF ایی که AccessViolation عمدی دارد: [WpfApplicationC00000005.zip](#)
به فایل‌های App.config و App.xaml.cs آن دقت کنید.
پروژه را کامپایل کرده و خارج از VS.NET اجرا کنید. خطا را نمایش می‌دهد ولی کرش نمی‌کند.

نویسنده: سوین
تاریخ: ۱۳۹۲/۱۲/۰۳ ۹:۲۱

با سلام
من تگ startup رو به صورت زیر نوشته بودم آیا می‌تونه تاثیر داشته باشه

```
<startup useLegacyV2RuntimeActivationPolicy="true">  
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>  
  <requiredRuntime version="v4.0.20506"/>  
</startup>
```

باز تست می‌کنم ببینم چی میشه ، انشالله که درست بشه . با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۳ ۹:۳۰

تنظیم یاد شده مربوط به تگ [runtime](#) است.