Subject: Assert failed in core/value.h line 464
Posted by jerson on Wed, 25 Aug 2010 06:29:04 GMT
View Forum Message <> Reply to Message

Quote:
From JFsrtsystem.2010-08-25-11-38-20.buglog file

* C:\upp2625\out\MINGW.Debug.Debug_full.Gui\JFsrtsystem.exe 25.08.2010 11:38:20, user: Jerson

ASSERT FAILED: Assertion failed in C:\upp2625\uppsrc\Core\Value.h, line 464
Invalid value conversion: N3Upp12RichValueRepINS_6StringEEE -> d

This is the message I get when I try to access the ExciterVoltage.Range member from the Settings structure.

```
// Global Settings go here
struct T_RGOP {
 Value  Range,
   Gain,
   Offset,
   Precision;

 void Serialize(Stream& stream)
 {
  stream % Range % Gain % Offset % Precision;
 };
};

// Speed/ Sensitivity enumerations
enum eSpeed {Zero=0, Low, Mid, High};

struct T_Settings {
// for system settings screen
 T_RGOP ExciterVoltage,  // settings for each type of parameter
   ExciterCurrent,
   OutputVoltage,
   OutputCurrent,
   InputVoltage,
   InputCurrent,
   RegulatorVoltage,
   RegulatorCurrent;
 Value ProtectFactor,
   ProtectMaxVout,
   ProtectMaxIout,
   ProtectMaxIin,
```

```
  RegV0, RegVmax,  // values for regulator 0 and 100%
  GapV0, GapVmax;  //     and gap 0 & 100%
int  ArcSensitivity;

// Other settings
int iHV_Mins,  iHV_Secs,
 iHV_Output, iHV_Protect,
 iRegRange,  iRegSpeed,
 iResGap,   iResSpeed;
int biAlarmSound:1,  // alarm sound on/off
 biAlarmLight:1;  // alarm light on/off

// default values for the settings
T_Settings()
{
 ExciterVoltage.Range = 6000;
 ExciterCurrent.Range = 2000;
 OutputVoltage.Range  = 400;
 OutputCurrent.Range  = 1000;
 InputVoltage.Range   = 430;
 InputCurrent.Range   = 500;
 RegulatorVoltage.Range = 1000;
 RegulatorCurrent.Range = 1000;
 ProtectFactor  = 1200;
 ProtectMaxVout = 1000;
 ProtectMaxIout = 1000;
 ProtectMaxIin = 1000;
 RegV0 = 0; RegVmax = 100;
 GapV0 = 0; GapVmax = 100;

 iHV_Mins = 1; iHV_Secs = 0;
}

// serialization function
void Serialize(Stream& stream)
{
 stream
 % ExciterVoltage % ExciterCurrent
 % OutputVoltage  % OutputCurrent
 % InputVoltage  % InputCurrent
 % RegulatorVoltage % RegulatorCurrent
 % ProtectFactor  % ProtectMaxVout
 % ProtectMaxIout % ProtectMaxIin
 % RegV0    % RegVmax
 % GapV0    % GapVmax
 % ArcSensitivity
 // Other settings
 % iHV_Mins % iHV_Secs
```

```
  % iHV_Output % iHV_Protect
  % iRegRange % iRegSpeed
  % iResGap % iResSpeed
/*  ("biAlarmSound", biAlarmSound)
  ("biAlarmLight", biAlarmLight)
*/
  ;
 }
};

T_Settings Settings;
```

This is the line of code that is driving me nuts. Maybe I am doing something wrong.
Vexciter.meter.SetMax(Settings.ExciterVoltage.Range);

If I comment out this line and put in a
DUMP(Settings.ExciterVoltage.Range);
I get the above assertion in both cases. Another place I have problems is using the same thing like this
Vexciter.meter.SetStep(Settings.ExciterVoltage.Range / 4); which comes back with an ambiguous divide operator message.

I am relatively new to C++ and UPP and cannot figure this out for myself. Can anyone guide me please?

Thanks
Jerson

Subject: Re: Assert failed in core/value.h line 464
Posted by koldo on Wed, 25 Aug 2010 09:35:15 GMT
View Forum Message <> Reply to Message

Hello Jerson

I do not understand well your problem. However I will put some ideas:

This code gives compiling errors because it does not know in advance the type of val:
Value val = 35;
double res = val/2;
This one works:
Value val = 35;
double res = int(val)/2.;
This one works too:
Value val = 35;
double res = double(val)/2.;
But this one will fail in runtime with an Assertion:

```
Value val = "35";
double res = double(val)/2.;
```
If you want to avoid an Assertion but your code could put an inappropriate value in Value, you can do this:
```
Value val = "35";
double res;
if (IsNumber(val))
 res = int(val)/2.;
else
 res  = -1; // You can put here and exception, an Exclamation, a return false, ...
```

Subject: Re: Assert failed in core/value.h line 464
Posted by jerson on Wed, 25 Aug 2010 11:17:49 GMT
View Forum Message <> Reply to Message

Hi Koldo

Thanks for taking a shot at the problem. It is none of those. I have zipped up a test case in my Controlsdemo file so you can experience the problem first hand.

The problem areas are in the main.cpp and are highlighted with capitalized comments.

It seems like the load from config file is causing the values to fail.

Regards

File Attachments
1) JFControls.7z, downloaded 226 times

Subject: Re: Assert failed in core/value.h line 464
Posted by jerson on Wed, 25 Aug 2010 14:49:58 GMT
View Forum Message <> Reply to Message

Ok, I found it

The Settings.Vexciter.Range parameter is of Value type.

The control that was being used to take in user input was of type EditString.

This was the assignment
Settings.Vexciter.Range = ~esExciterVoltage;
This is where things were going wrong. The String value was geting assigned to the variable.

Following this was a call to StoreToFile(Settings, cfgfile);

this was storing the value as a string in the cfgfile.

On reloading the file by using LoadFromFile(Settings, cfgfile), the string was being loaded to Value. On assigning this to Meter.SetMax(Settings.Vexciter.Range), I was getting the assertion since the Value does not seamlessly convert to a double(help file says so).

Doing this Meter.SetMax((double)Settings.Vexciter.Range) does not help either.

However, now, I am out of the block and hopefully can steam ahead. I have changed all the user input boxes to EditDouble type.

BTW: why does this work
```
double d = Settings.ExciterVoltage.Range;
Vexciter.meter.SetStep(d/4);
```

and why not this?
```
Vexciter.meter.SetStep(Settings.ExciterVoltage.Range/4);
```

this gives an ambiguous operator message and fails to compile

Thanks for your help

Subject: Re: Assert failed in core/value.h line 464
Posted by koldo on Wed, 25 Aug 2010 15:46:47 GMT
View Forum Message <> Reply to Message

Hello Jerson

Good for finding it.

Remember my first post:

This code gives compiling errors because the compiler does not know in advance the type of val:
```
Value val = 35;
double res = val/2;
```
This one works:
```
Value val = 35;
double res = int(val)/2.;
```
