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
