
Subject: Re: Function.h: "expression cannot be used as a function"

Posted by [Giorgio](#) on Thu, 01 Feb 2018 09:47:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Wed, 31 January 2018 18:46

The problem will be around here:

/home/pi/MyApps/StatusDetector/AggiungiComm.cpp:149:1:

Can you show us a couple of lines around it? Or maybe the whole file, if possible.

This is the whole file. The line numbers differ from those in the log because I removed the comments. In the original file the line number 149 is the last line of the file and is blank.

```
#include "AggiungiComm.h"
```

```
AggiungiComm::AggiungiComm(MLav _mlav, char tipo)
{
    String macchina = _mlav.GetMacchina();
    CtrlLayoutOKCancel(*this, t_("Selezionare commessa e pezzatura"));
    Add(strCommPezz.LeftPosZ(16, 148).TopPosZ(12, 19));
    btnTrova <=<= THISBACK1(ScanCommessa, macchina);
    listaMetri <=<= THISBACK(CambioPezzatura);
    DisableFields();
    strMacchina = macchina;
    ok.Disable();
    btnTrova.Ok();
    if (tipo == 'B')
    {
        ok.Enable();
        ok.Ok();
        btnTrova.Normal().Disable();
        strCommPezz <=<= AsString(_mlav.lav_in_corso.COMM) + AsString(_mlav.lav_in_corso.FASE)
+ "1" ;
        ScanCommessa(macchina);
    }
}
```

```
void AggiungiComm::ScanCommessa(String macchina)
{
    if (SplittaComm() > 0)
    {
        ok.Enable();
        ok.Ok();
        btnTrova.Normal().Disable();
    } else {
```

```

    PromptOK(t_("ATTENZIONE: commessa non trovata, effettuare di nuovo lo scan"));
    CleanFields();
    strMacchina = macchina;
}
}

```

```

void AggiungiComm::CleanFields()
{
    strLav3A <=<= Null;
    strLav3B <=<= Null;
    strLav3C <=<= Null;
    strLav3D <=<= Null;
    strLav3Dbis <=<= Null;
    strMacchina <=<= Null;
    strCommPezz <=<= Null;
    listaMetri.Reset();
    strCommPezz.SetFocus();
}

```

```

void AggiungiComm::DisableFields()
{
    strLav3A.Disable();
    strLav3B.Disable();
    strLav3C.Disable();
    strLav3D.Disable();
    strLav3Dbis.Disable();
    strMacchina.Disable();
    strCommPezz.SetFocus();
}

```

```

int AggiungiComm::SplittaComm()
{
    String ingresso = this->strCommPezz.GetData();
    WString pulita = (WString)TrimBoth(ingresso);
    pulita = pulita.Left(pulita.GetLength() - 1);
    WString comm = pulita.Left(pulita.GetLength() - 2);
    WString fas = pulita.Right(2);
    strLav3A = comm;
    strLav3B = fas;
    strLav3C = db_data.ContaBobFatte(AsString(comm), AsString(fas)) + 1;
    strLav3D = NULL;
    strLav3Dbis = NULL;
    int esiste_comm = db_data.ControllaCommessa(AsString(comm));
    if (esiste_comm > 0)
    {
        if (fas == (WString)"90")
        {
            InserisciPezzature(AsString(comm), AsString(fas));

```

```

}
else {
    vcodbob.resize(1);
    vmtbob.resize(1);
    std::fill(vcodbob.begin(), vcodbob.end(), "");
    std::fill(vmtbob.begin(), vmtbob.end(), 0);
    listaMetri = 0;
}
}
return esiste_comm;
}

```

```

void AggiungiComm::InserisciPezzature(String commessa, String fase)
{
    std::vector<PEZZE_BOB> lista = db_data.TrovaPezzature(commessa, fase);
    size_t dimensione = lista.size();
    vcodbob.resize(dimensione);
    vmtbob.resize(dimensione);
    int m = 0;
    for (size_t i=0; i<lista.size(); i++)
    {
        int num_bob = lista[i].PEZZE.N_BOB;
        int k = 0;
        for(int j = 0; j < num_bob; j++)
        {
            if (AsString(lista[i].PEZZE.BOB) == "777")
                listaMetri.Set(m, m, t_(" Pezz. non prevista"));
            else
                listaMetri.Set(m, m, ( AsString(lista[i].PEZZE.MT) + "m - N: " + AsString(k+1) + " di " +
                AsString(num_bob)));
            vcodbob[m] = lista[i].PEZZE.BOB;
            vmtbob[m] = lista[i].PEZZE.MT;
            if (k<lista[i].fatte) { listaMetri.DisableCase(m); }
            k++;
            m++;
        }
        k = 0;
    }
    listaMetri = m -1;
    strLav3D.MaxLen(3);
    strLav3D <=<= "777" ;
    strLav3Dbis = 0 ;
}

```

```

void AggiungiComm::CambioPezzatura()
{
    strLav3D.MaxLen(3);
    strLav3D <=<= vcodbob[listaMetri] ;
}

```

```
    strLav3Dbis = vmtbob[listaMetri] ;  
}
```
