
Subject: Re: weird reaction with const w_chart
Posted by [iRANian](#) on Fri, 09 Mar 2012 13:53:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

You're using %s with a wchar_t in formatted functions, you need to use %S. Casting a char to wchar_t (if it works) can cause memory corruption. Instead of using wchar_t, use StringClass. Instead of using Console_Input, use a wrapper function that takes formatted input like Console_Output() does. This is how I would write the code:

Toggle Spoiler// Console_Input() taking formatted input

```
void Console(const char *Format, ...)
```

```
{  
    char buffer[256];  
    va_list va;  
    _crt_va_start(va, Format);  
    vsnprintf(buffer, 256, Format, va);  
    va_end(va);  
    Console_Input(buffer);  
}
```

// Call Kambot_Commands() via chat

```
bool Kambot::OnChat(int PlayerID, TextMessageEnum Type, const wchar_t *Message, int  
recieverID)
```

```
{  
    StringClass Msg = Message;  
    Kambot_Commands(PlayerID, Type, Msg, recieverID);  
  
    return true;  
}
```

```
void Kambot_Commands(int PlayerID, TextMessageEnum Type, StringClass Msg, int recieverID)
```

```
{  
    Console("MSG Debug: Kambot_Commands() called"); // DEBUG CRAP  
    Console("MSG Debug: Kambot_Commands() MSG == %s", Msg); // DEBUG CRAP  
    if (Msg[0] == '!')  
    {  
        Console("MSG Debug: Kambot_Commands() Command triggered"); // DEBUG CRAP  
    }  
}
```

// Call Kambot_Commands() via a Keyhook (using keys.cfg keys)

```
void KB_keyhook::KeyHook()
```

```
{  
    if((The_Game()->Get_Game_Duration_S() - LastPress) >= 1)  
    {  
        StringClass Msg = Get_Parameter("Command");
```

```
        Kambot_Commands(Get_Player_ID(Owner()), TEXT_MESSAGE_TEAM, Msg, -1); // Last
```

parameter isn't used

```
Console("MSG Debug: KeyHook() called with %s", Msg); // DEBUG CRAP
```

```
LastPress = The_Game()->Get_Game_Duration_S();  
}  
}
```

Not sure if it actually runs correctly though, didn't bother checking. You can also use the `__FUNCTION__` and `__LINE__` macros to grab the function the code is executing and the line number while debugging.
