Subject: analizing a crash file
Posted by forlano on Tue, 29 May 2007 21:58:57 GMT
View Forum Message <> Reply to Message

Hello,

I've built the crash package to debug my app. I've even produced a map file. Now the program show me a lot of lines with many interesting data that unfortunately are without meaning for me (see below for few lines).
The question is: what I must look for in these lines to understand where something goes wrong? It seems there is a problem in the function SwissDubov::CanPairingGroup.

Many thanks,
Luigi

==========
Crash log VegaTeam5.exe.2007-05-29-23-42-56.crash
Access violation reading at 0x00133000
0x00435057: int SwissDubov::CanPairingGroup(int,int,int,int (* const)[2],int *) + 0x554 bytes

Recognized stack dwords:
0x00512e09: _$I10_OUTPUT + 0x88d bytes
0x0050f91a: __fltout2 + 0x64 bytes
0x0050086a: __shift + 0x18 bytes
0x004ffc96: __getptd_noexit + 0x7e bytes
0x005080ee: _write_string + 0x27 bytes
0x005089ff: __output_l + 0x8ee bytes
0x00508a7e: __output_l + 0x96d bytes
0x005010f9: __cftog_l + 0xf6 bytes
0x004ffc96: __getptd_noexit + 0x7e bytes
0x005080ee: _write_string + 0x27 bytes
0x005089ff: __output_l + 0x8ee bytes
0x00508a7e: __output_l + 0x96d bytes
0x004ffc96: __getptd_noexit + 0x7e bytes
0x005080ee: _write_string + 0x27 bytes
0x005089ff: __output_l + 0x8ee bytes
0x00508a7e: __output_l + 0x96d bytes
0x004ffc96: __getptd_noexit + 0x7e bytes
0x005080ee: _write_string + 0x27 bytes
0x005089ff: __output_l + 0x8ee bytes
0x00508a7e: __output_l + 0x96d bytes
0x004fba7c: _free + 0x6e bytes
0x004fba9b: _free + 0x8d bytes
0x004fba7c: _free + 0x6e bytes
0x004fba9b: _free + 0x8d bytes
0x005003e0: __except_handler4 + 0x0 bytes
0x004fba9b: _free + 0x8d bytes
0x004cc65c: void Upp::MemoryFree(void *) + 0x3c bytes

0x004d8f2f: class Upp::String Upp::VFormat(char const *,char *) + 0x9f bytes
0x0052b9d1: __ehhandler$?VFormat@Upp@@YA?AVString@1>@PBDPAD>@Z + 0x0 bytes
0x004d91de: class Upp::String Upp::Sprintf(char const *,...) + 0x1e bytes
0x004d954c: class Upp::String Upp::FormatInteger(int) + 0x4c bytes
0x004d9568: class Upp::String Upp::FormatInteger(int) + 0x68 bytes
0x004d9588: class Upp::String Upp::FormatInteger(int) + 0x88 bytes
0x0052bb1b: __ehhandler$?FormatInteger@Upp@@YA?AVString@1>@H>@Z + 0x0 bytes
0x004d688f: void Upp::Stream::Putf(class Upp::String const &) + 0x1f bytes
0x004cd672: Upp::String::~String(void) + 0x12 bytes
0x00418e04: ??$?6H@Upp@@YAAAVStream>@0>@AAV10>@ABH>@Z + 0x32 bytes
0x004d5f6b: void Upp::Stream::Putf(char const *) + 0x1b bytes
0x0043066b: void SwissDubov::MovePlayer(int,int,int,int,int) + 0x1c9 bytes
0x0051ac61: __ehhandler$?MovePlayer@SwissDubov@@QAEXHHHHH>@Z + 0x0 bytes
0x00430c6b: int SwissDubov::MergeGroup(int) + 0x1b1 bytes
0x00430f1b: int SwissDubov::MergeGroup(int) + 0x461 bytes
0x004fba7c: _free + 0x6e bytes
0x004fba9b: _free + 0x8d bytes
0x005003e0: __except_handler4 + 0x0 bytes
0x004fba9b: _free + 0x8d bytes
0x004cc65c: void Upp::MemoryFree(void *) + 0x3c bytes
0x004d8f2f: class Upp::String Upp::VFormat(char const *,char *) + 0x9f bytes
0x0052b9d1: __ehhandler$?VFormat@Upp@@YA?AVString@1>@PBDPAD>@Z + 0x0 bytes
0x004d91de: class Upp::String Upp::Sprintf(char const *,...) + 0x1e bytes
0x004d954c: class Upp::String Upp::FormatInteger(int) + 0x4c bytes
0x004d9568: class Upp::String Upp::FormatInteger(int) + 0x68 bytes
0x004d9588: class Upp::String Upp::FormatInteger(int) + 0x88 bytes
0x0052bb1b: __ehhandler$?FormatInteger@Upp@@YA?AVString@1>@H>@Z + 0x0 bytes
0x0051b361: __ehhandler$?CanPairingGroup@SwissDubov@@QAEHHHHQAY01HPAH>@Z + 0x0 bytes
0x00435ec1: int SwissDubov::PrepareRound(int,int) + 0x27b bytes
0x004ffc96: __getptd_noexit + 0x7e bytes
0x005080ee: _write_string + 0x27 bytes
0x005089ff: __output_l + 0x8ee bytes
0x00508a7e: __output_l + 0x96d bytes
0x004fba7c: _free + 0x6e bytes
0x004fba9b: _free + 0x8d bytes
0x005003e0: __except_handler4 + 0x0 bytes
0x004fba9b: _free + 0x8d bytes
0x004cc65c: void Upp::MemoryFree(void *) + 0x3c bytes
0x004d8f2f: class Upp::String Upp::VFormat(char const *,char *) + 0x9f bytes
0x0052b9d1: __ehhandler$?VFormat@Upp@@YA?AVString@1>@PBDPAD>@Z + 0x0 bytes
0x004d91de: class Upp::String Upp::Sprintf(char const *,...) + 0x1e bytes
0x004023ed: int Upp::AIndex<class Upp::String,class Upp::Vector<class Upp::String>,struct Upp::StdHash<class Upp::String> >::Find0(class Upp::String const &,int)const + 0x19 bytes
0x004027d0: int Upp::AIndex<class Upp::String,class Upp::Vector<class Upp::String>,struct Upp::StdHash<class Upp::String> >::Find(class Upp::String const &,unsigned int)const + 0x1b bytes
0x0040292b: int Upp::AIndex<class Upp::String,class Upp::Vector<class Upp::String>,struct

Upp::StdHash<class Upp::String> >::Find(class Upp::String const &)const + 0x1e bytes
 0x004081a2: class TeamArchive & Upp::AMap<class Upp::String,class TeamArchive,class Upp::Vector<class TeamArchive>,struct Upp::StdHash<class Upp::String> >::Get(class Upp::String const &) + 0xc bytes
 0x004cd672: Upp::String::~String(void) + 0x12 bytes
 0x0042c159: void RoundData::AdjournTeamState(int) + 0x50a bytes
 0x0051b3c8: __ehhandler$?PrepareRound@SwissDubov@@QAEHHH>@Z + 0x0 bytes
 0x0042df67: int RoundData::MakeNextPairing(void) + 0x196 bytes
 0x004cc993: unsigned char * Upp::Alloc4KBRaw(void) + 0xb3 bytes
 0x004cc993: unsigned char * Upp::Alloc4KBRaw(void) + 0xb3 bytes
 0x004cc993: unsigned char * Upp::Alloc4KBRaw(void) + 0xb3 bytes
 0x004b4040: long Upp::Ctrl::WindowProc(unsigned int,unsigned int,long) + 0xba3 bytes
 0x004b40ae: long Upp::Ctrl::WindowProc(unsigned int,unsigned int,long) + 0xc11 bytes
 0x004d3f8e: Upp::PtrBase::~PtrBase(void) + 0x1e bytes
 0x004b7252: void Upp::Vector<struct Upp::Ctrl::MoveCtrl>::Trim(int) + 0x20 bytes
 0x004b8424: void Upp::AIndex<class Upp::Ctrl *,class Upp::Vector<class Upp::Ctrl *>,struct Upp::StdHash<class Upp::Ctrl *> >::Clear(void) + 0x15 bytes
 0x004b3069: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x17e bytes
 0x004b4040: long Upp::Ctrl::WindowProc(unsigned int,unsigned int,long) + 0xba3 bytes
 0x004b40ae: long Upp::Ctrl::WindowProc(unsigned int,unsigned int,long) + 0xc11 bytes
 0x004b72de: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x82 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b1741: bool Upp::Ctrl::IsWndOpen(void)const + 0x5 bytes
 0x004b0058: bool Upp::Ctrl::IsOpen(void)const + 0xf bytes
 0x004a6a98: struct Upp::Rect_<int> Upp::Rect_<int>::Inflated(int)const + 0xd bytes
 0x004b785a: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0x10 bytes
 0x004b1741: bool Upp::Ctrl::IsWndOpen(void)const + 0x5 bytes
 0x004b0058: bool Upp::Ctrl::IsOpen(void)const + 0xf bytes
 0x004b1741: bool Upp::Ctrl::IsWndOpen(void)const + 0x5 bytes
 0x0047ae5b: Upp::WithFileSelectorLayout<class Upp::TopWindow>::~WithFileSelectorLayout<class Upp::TopWindow>(void) + 0xf1 bytes
 0x0052395c: __ehhandler$??1?$WithFileSelectorLayout@VTopWindow>@Upp@@@Upp@@UAE>@XZ + 0x0 bytes
 0x0052f638: ??__Fx@?7??sStdColor@Upp@@YAAAV?$Vector@K>@1>@XZ>@YAXXZ + 0xbb8 bytes
 0x0052f624: ??__Fx@?7??sStdColor@Upp@@YAAAV?$Vector@K>@1>@XZ>@YAXXZ + 0xba4 bytes
 0x0052f638: ??__Fx@?7??sStdColor@Upp@@YAAAV?$Vector@K>@1>@XZ>@YAXXZ + 0xbb8 bytes
 0x0052f638: ??__Fx@?7??sStdColor@Upp@@YAAAV?$Vector@K>@1>@XZ>@YAXXZ + 0xbb8 bytes
 0x00512e09: _$I10_OUTPUT + 0x88d bytes
 0x0050f91a: __fltout2 + 0x64 bytes
 0x0050086a: __shift + 0x18 bytes
 0x004ffc96: __getptd_noexit + 0x7e bytes
 0x005089ff: __output_l + 0x8ee bytes

```
 0x00508a7e: __output_l + 0x96d bytes
 0x004c40f0: class Upp::FontInfo Upp::Draw::Acquire(class Upp::Font,struct HDC__ *,int,int) +
0x9b bytes
 0x004bbb30: void Upp::Draw::SetFont(class Upp::Font,int) + 0x7b bytes
 0x004c36e2: void Upp::Draw::DrawTextOp(int,int,int,unsigned short const *,class Upp::Font,class
Upp::Color,int,int const *) + 0x83 bytes
 0x004c40f0: class Upp::FontInfo Upp::Draw::Acquire(class Upp::Font,struct HDC__ *,int,int) +
0x9b bytes
 0x004c405f: class Upp::FontInfo Upp::Draw::Acquire(class Upp::Font,struct HDC__ *,int,int) +
0xa bytes
 0x004b1db2: void Upp::Ctrl::WndInvalidateRect(struct Upp::Rect_<int> const &) + 0x16 bytes
 0x004b7907: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xbd bytes
 0x004b72de: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x82 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b78fd: void Upp::Ctrl::RefreshFrame(struct Upp::Rect_<int> const &) + 0xb3 bytes
 0x004b72e9: void Upp::Ctrl::Refresh(struct Upp::Rect_<int> const &) + 0x8d bytes
 0x0051ab8c: __ehhandler$?MakeNextPairing@RoundData@@QAEHXZ + 0x0 bytes
 0x00409c08: void VegaMain::DopairingCB(void) + 0xc8 bytes
 0x00409b4a: void VegaMain::DopairingCB(void) + 0xa bytes
 0x00515c8a: __ehhandler$?DopairingCB@VegaMain@@QAEXXZ + 0x0 bytes
 0x00460fc8: bool Upp::Pusher::FinishPush(void) + 0x25 bytes
 0x004619ac: void Upp::Pusher::LeftUp(struct Upp::Point_<int>,unsigned long) + 0x5 bytes
 0x004b4a18: class Upp::Image Upp::Ctrl::MouseEvent(int,struct Upp::Point_<int>,int,unsigned
long) + 0x1e3 bytes
 0x004b528e: class Upp::Image Upp::Ctrl::MouseEventH(int,struct Upp::Point_<int>,int,unsigned
long) + 0xc4 bytes
 0x004b5556: class Upp::Image Upp::Ctrl::MEvent0(int,struct Upp::Point_<int>,int) + 0x2b3 bytes
 0x00528beb:
__ehhandler$?MEvent0@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>@H>@Z +
0x0 bytes
 0x004b5971: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x199 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
```

```
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b594e: class Upp::Image Upp::Ctrl::DispatchMouseEvent(int,struct Upp::Point_<int>,int) +
0x176 bytes
 0x004b8d97: Upp::TimeEvent::~TimeEvent(void) + 0x26 bytes
 0x00528f8c: __ehhandler$??1TimeEvent@Upp@@QAE>@XZ + 0x0 bytes
 0x00528c64:
__ehhandler$?DispatchMouseEvent@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>
@H>@Z + 0x0 bytes
 0x004b5c46: class Upp::Image Upp::Ctrl::DispatchMouse(int,struct Upp::Point_<int>,int) + 0xd5
bytes
 0x00528ca2:
__ehhandler$?DispatchMouse@Ctrl>@Upp@@AAE?AVImage@2>@HU?$Point_@H@2>@H>
@Z + 0x0 bytes
 0x004b1c12: class Upp::Image Upp::Ctrl::DoMouse(int,struct Upp::Point_<int>,int) + 0x43 bytes
```

0x005288c0: __ehhandler$?DoMouse@Ctrl>@Upp@@IAE?AVImage@2>@HU?$Point_@H@2>@H>@Z + 0x0 bytes
0x004b3e47: long Upp::Ctrl::WindowProc(unsigned int,unsigned int,long) + 0x9aa bytes
0x004ae242: struct Upp::Rect_<int> Upp::Ctrl::GetScreenView(void)const + 0x34 bytes
0x00471ddc: struct Upp::Rect_<int> Upp::operator+(struct Upp::Rect_<int>,struct Upp::Point_<int>) + 0x21 bytes
0x004ae2cb: struct Upp::Rect_<int> Upp::Ctrl::GetScreenRect(void)const + 0x61 bytes
0x004add82: struct Upp::Rect_<int> Upp::Ctrl::GetRect(void)const + 0xd bytes
0x004ae27e: struct Upp::Rect_<int> Upp::Ctrl::GetScreenRect(void)const + 0x14 bytes
0x004b34ab: long Upp::Ctrl::WindowProc(unsigned int,unsigned int,long) + 0xe bytes
0x00528b60: __ehhandler$?WindowProc@Ctrl>@Upp@@UAEJIIJ>@Z + 0x0 bytes
0x00471ddc: struct Upp::Rect_<int> Upp::operator+(struct Upp::Rect_<int>,struct Upp::Point_<int>) + 0x21 bytes
0x004ae242: struct Upp::Rect_<int> Upp::Ctrl::GetScreenView(void)const + 0x34 bytes
0x00471ddc: struct Upp::Rect_<int> Upp::operator+(struct Upp::Rect_<int>,struct Upp::Point_<int>) + 0x21 bytes
0x004ae2cb: struct Upp::Rect_<int> Upp::Ctrl::GetScreenRect(void)const + 0x61 bytes
0x004add82: struct Upp::Rect_<int> Upp::Ctrl::GetRect(void)const + 0xd bytes
0x004ae27e: struct Upp::Rect_<int> Upp::Ctrl::GetScreenRect(void)const + 0x14 bytes
0x004b41e8: long Upp::TopWindow::WindowProc(unsigned int,unsigned int,long) + 0x91 bytes
0x004b8424: void Upp::AIndex<class Upp::Ctrl *,class Upp::Vector<class Upp::Ctrl *>,struct Upp::StdHash<class Upp::Ctrl *> >::Clear(void) + 0x15 bytes
0x004b34ab: long Upp::Ctrl::WindowProc(unsigned int,unsigned int,long) + 0xe bytes
0x00528b60: __ehhandler$?WindowProc@Ctrl>@Upp@@UAEJIIJ>@Z + 0x0 bytes
0x004b3069: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x17e bytes
0x004b2eeb: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x0 bytes
0x004b2eeb: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x0 bytes
0x004add82: struct Upp::Rect_<int> Upp::Ctrl::GetRect(void)const + 0xd bytes
0x004b41e8: long Upp::TopWindow::WindowProc(unsigned int,unsigned int,long) + 0x91 bytes
0x0046aa77: Upp::Ptr<class Upp::Ctrl>::Ptr<class Upp::Ctrl>(class Upp::Ctrl *) + 0x29 bytes
0x004b302c: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x141 bytes
0x004b2eeb: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x0 bytes
0x00528b43: __ehhandler$?WndProc@Ctrl>@Upp@@KGJPAUHWND__@@IIJ>@Z + 0x0 bytes
0x004b2eeb: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x0 bytes
0x004b2eeb: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x0 bytes
0x004b2eeb: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x0 bytes
0x004b2eeb: long Upp::Ctrl::WndProc(struct HWND__ *,unsigned int,unsigned int,long) + 0x0 bytes

0x004b1cd5: bool Upp::Ctrl::ProcessEvent(bool *) + 0xa1 bytes
0x004b1cf2: bool Upp::Ctrl::ProcessEvents(bool *) + 0xa bytes
0x004b30db: void Upp::Ctrl::EventLoop(class Upp::Ctrl *) + 0x39 bytes
0x004b2c41: class Upp::Ctrl * Upp::Ctrl::GetOwner(void) + 0x5 bytes
0x004afa9e: int Upp::TopWindow::Run(bool) + 0x113 bytes
0x005284ac: __ehhandler$?Run@TopWindow>@Upp@@QAEH_N@Z + 0x0 bytes
.... [continues....]

Subject: Re: analizing a crash file
Posted by mirek on Wed, 30 May 2007 09:13:38 GMT
View Forum Message <> Reply to Message

.crash results do not always give the precise information, I am afraid this is one of such cases.
The only conclusion you can take here is that really crashes in CanPairingGroup...

Does it crash in debug mode too?

Subject: Re: analizing a crash file
Posted by forlano on Wed, 30 May 2007 11:32:44 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 30 May 2007 11:13.crash results do not always give the precise information, I am afraid this is one of such cases. The only conclusion you can take here is that really crashes in CanPairingGroup...

Does it crash in debug mode too?

Thanks,

this is at least something.
In debug mode does not crash. I'm going to investigate CanPairingGroup...

Luigi

Subject: Re: analizing a crash file
Posted by forlano on Wed, 30 May 2007 15:27:26 GMT
View Forum Message <> Reply to Message

Fixed!

It was the classic unitialized variable .
Perhaps in debug mode they are set to 0. Anyway the crash tool put me in the right direction.
Thanks,

Luigi

Subject: Re: analizing a crash file
Posted by mirek on Wed, 30 May 2007 15:57:53 GMT
View Forum Message <> Reply to Message

forlano wrote on Wed, 30 May 2007 11:27Fixed!

It was the classic unitialized variable .
Perhaps in debug mode they are set to 0. Anyway the crash tool put me in the right direction.
Thanks,

Luigi

IME, if release crashes and debug not, it is 80% unitialized variable, 10% compiler bug....