Subject: Re: Problems debugging with Visual C++
Posted by mirek on Thu, 18 Sep 2014 10:55:30 GMT
View Forum Message <> Reply to Message

Interesting. May I ask for more logs?

```
uint64 Pdb::GetCpuRegister(const Context& ctx, int sym)
{
 int q = GetRegisterList().Find(sym);
 RLOG("== GetCpuRegister =========");
 RDUMP(sym);
 RDUMP(q);
 if(q < 0)
  return 0;
 const CpuRegister& r = GetRegisterList()[q];
 RDUMP(r.name);
 RDUMP(r.kind);
 RDUMP(r.sym);
#ifdef CPU_64
 uint64 val = win64 ? GetRegister64(ctx, sym) : GetRegister32(ctx, sym);
#else
 uint64 val = GetRegister32(ctx, sym);
#endif
 RDUMP(Format64Hex(val));
 switch(r.kind) {
 case REG_L:
  return LOBYTE(val);
 case REG_H:
  return HIBYTE(val);
 case REG_X:
  return LOWORD(val);
 case REG_E:
  return LODWORD(val);
 }
 return val;
}
```

```
BOOL CALLBACK Pdb::EnumLocals(PSYMBOL_INFO pSym, ULONG SymbolSize, PVOID
UserContext)
{
 LocalsCtx& c = *(LocalsCtx *)UserContext;

 if(pSym->Tag == SymTagFunction)
  return TRUE;
```

```
 RLOG("=== EnumLocals ========");
 RDUMP(UserContext);
 RDUMP(c.context);
 RDUMP(pSym->Register);
 Val& v = (pSym->Flags & IMAGEHLP_SYMBOL_INFO_PARAMETER ? c.param :
c.local).GetAdd(pSym->Name);
 v.address = (adr_t)pSym->Address;
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGISTER)
  v.address = pSym->Register;
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE) {
  if(pSym->Register == CV_ALLREG_VFRAME) {
  #ifdef CPU_64
   if(c.pdb->win64)
    v.address += c.pdb->GetCpuRegister(*c.context, CV_AMD64_RBP);
   else
  #endif
    v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, CV_REG_EBP);
  }
  else
   v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, pSym->Register);
 }
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_FRAMERELATIVE)
  v.address += c.frame;
 c.pdb->TypeVal(v, pSym->TypeIndex, (adr_t)pSym->ModBase);
 LLOG("LOCAL " << pSym->Name << ": " << Format64Hex(v.address));
 return TRUE;
}
```

Thanks a lot, I feel we are really very close now.

Mirek
