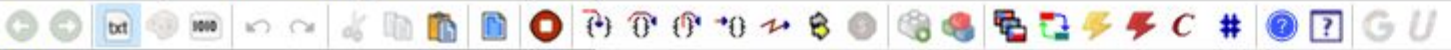

Subject: theide: New Threads tab in PDB debugger
Posted by [mirek](#) on Wed, 15 Jul 2020 11:37:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Should simplify investigation of complicated multithreaded issues:

File Attachments

1) [Clipboard01.jpg](#), downloaded 584 times



GUI

- plugin/wav
- Core**
- Core/SSL
- CtrlCore
- CtrlLib
- Draw
- Painter
- PdfDraw
- RichText
- plugin/bmp
- plugin/jpg
- plugin/png
- plugin/z
- <prj-aux>
- <ide-aux>
- <temp-aux>
- <meta>

- # Core.h
- # config.h
- # Defs.h
- # Ops.h
- # Fn.h
- Cpu.cpp
- # Mem.h
- C Mem.cpp
- # SIMD_SSE2.h
- C SIMD.cpp
- # Atomic.h
- # Mt.h
- C Mt.cpp
- # Heap.h
- # HeapImp.h
- C heaputil.cpp
- C hheap.cpp
- C sheap.cpp
- C lheap.cpp
- C heap.cpp
- C heapdbg.cpp
- # String.h
- # AString.hpp
- C StringFind.cpp
- C String.cpp
- C WString.cpp

- C SplitMerge.cpp
- CharSet.i
- # CharSet.h
- # Utf.hpp
- C Utf.cpp
- C UnicodeInfo.cpp
- C CharSet.cpp
- C Bom.cpp
- # Path.h
- C Path.cpp
- C NetNode.cpp
- # App.h
- C App.cpp
- # Huge.h
- C Huge.cpp
- # Stream.h
- C Stream.cpp
- C BlockStream.cpp
- C FileMapping.cpp
- # FilterStream.h
- C FilterStream.cpp
- # Profile.h
- # Diag.h
- C Log.cpp
- C Debug.cpp
- # Util.h

```

pp  AlphaBlend.h  Other.h  Stream.cpp  Stream.h  String.cpp

void AppExecute__(void (*app) ())
{
    try {
        (*app) ();
    }
    catch(ExitExc) {
        return;
    }
}

#ifdef PLATFORM_POSIX

void s_ill_handler(int)
{
    Panic("Illegal instruction!");
}

void s_segv_handler(int)
{
    Panic("Invalid memory access!");
}

void s_fpe_handler(int)
{
    Panic("Invalid arithmetic operation!");
}

void AppInit__(int argc, const char **argv, const char **
{
    SetLanguage(LNG_ENGLISH);
    sSetArgv0__(argv[0]);
    for(const char *var; (var = *envptr) != 0; envptr++)
    {

```

Autos Locals this Watches CPU Memory Threads **0x44f8**

- 0x44f8
 - GuiMainFn_()
 - Upp::AppExecute__(app=1400015f0)
 - WinMain(hInstance=140000000, lpCmdLine=188fb8e "", nCmdShow=10)
 - WinMainCRTStartup()
 - WinMainCRTStartup()
 - BaseThreadInitThunk()
 - RtlUserThreadStart()
- 0x39b0
- 0x46a4
 - NtWaitForWorkViaWorkerFactory()
 - RtlInitializeResource()
 - BaseThreadInitThunk()
 - RtlUserThreadStart()
- 0x26c8