

# XStudio 调试器协议 1.0

文档库地址 <https://xlang.link/documents/index.html>

(此文档适用的 XStudio 最低版本为 v1.0.18.7768)

XStudio 的调试器数据协议是为了使集成开发环境与程序调试器更好的结合，以达到可视化调试的目的而定制。

XStudio 与调试器之间使用二进制数据流的方式进行通讯。此协议适用于本地调试、远程互联网调试以及串口等通信方式的调试操作。

基本报文数据格式：

协议头	命令(Command)	上下文(线程 ID)	正文长度	正文
2 字节	4 字节	8 字节	4 字节	正文长度所描述的长度数据
0xFF 0xFE	Int 整数	Long 值	Int 整数	报文正文
固定标志	见 Command 附表	/	/	Json 结构正文

XRegular 表达式如下：

`[FFFE]{c:LEBB}<len:2>{t:LEBB}<len:8>{l:LEBB}<len:4>*<len:l>`

Command 列表：

名称	值	XStudio 发出说明	调试器发出说明	示例（参考开源代码）
Unknow	0	保留	/	
Interrupt	1	中断正在运行的线程\程序	被调试程序已中断	
Continue	2	使已中断的程序继续运行	/	
StepIn	3	源码级单步步入		
StepOver	4	源码级单步步过		
StepOut	5	源码级单步步出		
SetBreakPoint	6	设置断点	断点处理结果	
SetMemoryBreakPoint	7	设置内存断点		
SetFunctionBreakPoint	8	设置函数断点		
QueryStackStruct	9	查询栈结构		
QueryStackObject	10	查询栈对象		
QueryStackObjectDetails	11	查询栈对象的详细信息		

QueryHeapStruct	12	查询堆结构		
QueryHeapObject	13	查询堆对象		
QueryHeapObjectDetails	14	查询堆对象的详细信息		
QueryThreadCount	15	查询线程数量		
QueryStackFrame	16	查询栈帧		
SwitchThread	17	切换线程		
QueryFrame	18	查询帧	查询结果	
QueryMemoryInfo	19	查询内存信息	查询结果	
GC	20	立刻执行垃圾回收		
Log	21		打印日志到日志窗口	
TriggeBreakPoint	22		断点被触发	
WatchObject	23	添加监视对象	监事对象更新信息	
Active	24	激活调试器	调试器已激活	
ExceptionInterrput	25	异常中断		
Debug	26	打印信息到输出窗口		
QueryObject	27	查询对象信息	查询结果	
GcDump	28		Dump gc 信息	
QueryByName	29	根据名称查找对象	查询结果	
ModuleLoaded	30	/	模块已加载	
LookupMemory	31	查看内存	查询结果	
DumpMemory	32	转储内存	查询结果	
GCInfo	33		运行时 Gc 信息状态输出	
DBGExit	34	退出调试器	调试器退出	
MessageBox	35		在 XStudio 弹出消息框显示信息	
MessageBoxReply	36	消息确认结果		
QueryInstruction	37	查询指令		
InstructionShow	38	显示指令		
InstructionStepIn	39	指令级步入		
InstructionStepOver	40	指令级步过		
InstructionStepOut	41	指令级步出		
ModuleUnloaded	42	模块卸载		

参考资料及开源代码:

- 1.XStudio 项目的 XDebugger.xcsm 文件(IDE 端)
- 2.CDE(c/c++开发扩展)项目中的 GDBShell.x 文件(基于 GDB 的,对 C/C++可视化调试扩展)。

有疑问请进入  QQ 群: [591392649](https://jq.qq.com/?_w=1027&t=591392649), 进行交流探讨.

