Tbridge客户端用户手册

**版本信息**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本** | **更改日期** | **更改要点说明** | **编制** | **审核** | **批准** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 客户端简介 5](#_Toc340137174)

[2. TBC界面与功能介绍 5](#_Toc340137175)

[2.1. 主界面 5](#_Toc340137176)

[2.2. 地址栏 5](#_Toc340137177)

[2.3. 数据工具栏 6](#_Toc340137178)

[2.4. 过滤工具栏 7](#_Toc340137179)

[2.5. 数据列表视图 9](#_Toc340137180)

[2.6. 协议树视图 11](#_Toc340137181)

[2.7. 二进制数据视图 13](#_Toc340137182)

[2.8. XML数据视图 14](#_Toc340137183)

[2.9. 日志 14](#_Toc340137184)

[2.10. 统计 15](#_Toc340137185)

[2.11. 状态栏 15](#_Toc340137186)

[3. 配置文件 15](#_Toc340137187)

[3.1. 关键配置项 16](#_Toc340137188)

[4. 过滤条件 18](#_Toc340137189)

[5. 录制与回放 19](#_Toc340137190)

[5.1. 如何录制数据 19](#_Toc340137191)

[5.2. 如何回放数据 19](#_Toc340137192)

[5.3. OP指令 20](#_Toc340137193)

[5.4. TAG列 22](#_Toc340137194)

[5.5. UP与DOWN的对应关系 22](#_Toc340137195)

[6. 管理面板 23](#_Toc340137196)

[6.1. 过滤表达式管理面板 23](#_Toc340137197)

[6.2. 用例管理面板 24](#_Toc340137198)

[6.3. 服务器配置管理面板 25](#_Toc340137199)

[7. FAQ 26](#_Toc340137200)

[7.1. 如何在数据列表中添加协议树中某个Entry节点的预览？ 26](#_Toc340137201)

[7.2. 如何清空已经添加到数据包的Action操作？ 26](#_Toc340137202)

[7.3. 能不能在上下行数据包使用的MetaName一致时切换数据包的方向？ 26](#_Toc340137203)

[7.4. 我的游戏协议中有很多合包，有木有办法自动拆开合包呢？ 26](#_Toc340137204)

[7.5. 如何自动将满足条件的数据包标记为待发送数据？ 26](#_Toc340137205)

[7.6. 登陆时TBC提示找不到tdr文件怎么办？ 27](#_Toc340137206)

[7.7. 我确定TBS、游戏客户端和服务器都正常，但TBC就是没反应怎么办？ 27](#_Toc340137207)

[7.8. 为什么有些快捷方式不能用？ 27](#_Toc340137208)

[7.9. 如何查看自定义列的全路径？ 27](#_Toc340137209)

[7.10. 可不可以隐藏一些我不关注的固定列，比如GameDataLen？ 28](#_Toc340137210)

[7.11. 我在编写OP指令时经常要手动编写期望字段的路径，有木有简单一点的办法？ 28](#_Toc340137211)

[7.12. 我希望TBC能在我F5发送当前行数据后，立即选中下一行数据，这个支持么？ 28](#_Toc340137212)

[7.13. 我修改了一些XML配置，但又不想重启TBC，可以Reload配置么？ 28](#_Toc340137213)

[7.14. 出现了如下的异常时怎么办？ 28](#_Toc340137214)

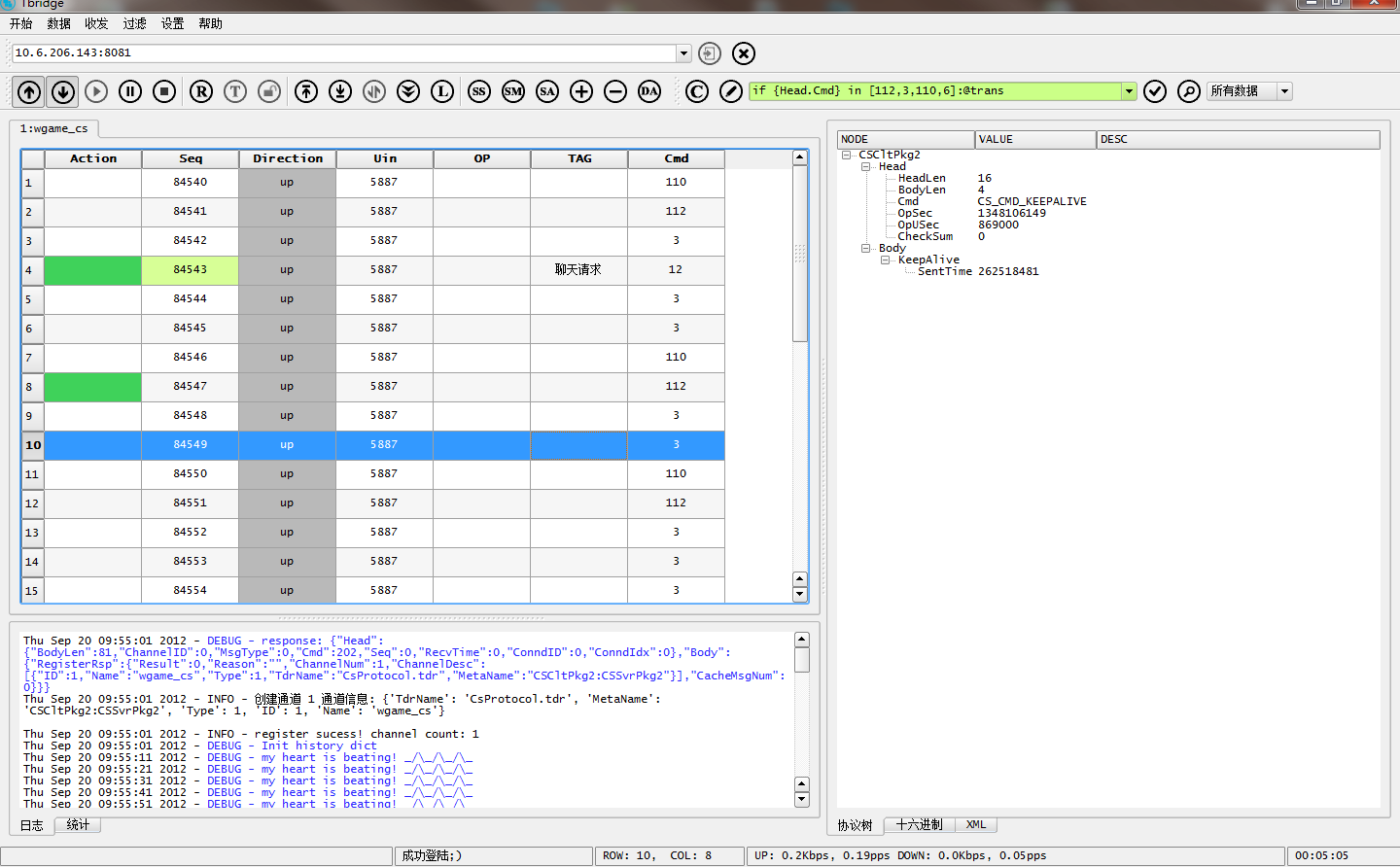
客户端简介

tbridge客户端通过TCP协议与tbridge服务器建立长连接，接收并展示服务器截取的上下行数据包。用户利用tbridge客户端提供的四种数据视图（列表、协议树、二进制、XML）可以非常方便地对数据包进行修改和重新打包。另外tbridge客户端提供“多通道”协议测试的支持，可以方便用户在同一界面的不同选项卡对多个C/S或S/S之间的协议交互进行测试。

为便于叙述，以下简称tbridge客户端为TBC，tbridge服务器为TBS。

TBC界面与功能介绍

* 1. 主界面



* 1. 地址栏



地址栏采用*IP:PORT?参数*格式。其中*IP*为TBS的监听地址，*参数*符合URL参数标准。当前仅支持gip、uin和cname参数。

其中gip用于指示TBC监听哪个主机上的游戏客户端。若不写gip则监听TBC所在主机上的游戏客户端数据。若gip=0.0.0.0则以观察者模式监听所有其它已登陆TBC上的游戏数据。

uin用于指示TBC监听使用哪个帐号登陆的游戏客户端，当同时使用gip和uin参数，且这时uin指定的帐号并未在gip指定的主机上登陆时，TBC会忽视gip设置，而以uin为准。

cname用于监听SS模式下的通道数据。例如在Wgame游戏中，我们配置了zoneSvr到mailSvr的名为wgame\_mail2zone的SS通道，则可在TBC使用cname=wgame\_mail2zone的URL参数来监听该通道上的数据。

*TIPS：观察者模式下TBS将不处理TBC发送的任何数据。不推荐同时使用gip和uin参数。使用cname监听SS通道数据时不可多人同时监听同一通道。*

*若需监听多个帐号或通道，则使用英文格式下的逗号间隔，例如：*

*10.6.206.143:uin=1234,5678*

*10.6.206.143:cname=wgame\_ss,wgame\_ss2*

* 1. 数据工具栏



该工具栏提供了数据相关的快捷按钮。

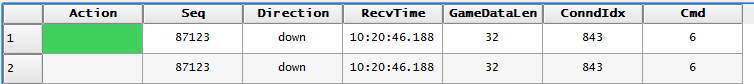
2.3.1、 按钮组中，按下按钮表示接收上行数据（游戏客户端到游戏服务器），按下按钮表示接收下行数据（游戏服务器到游戏客户端）。弹起按钮表示不接收对应方向的数据，如表示不接收上行数据。不接收的数据将直接从TBC透传回TBS。

2.3.2、 按钮组中，从左至右依次为继续（F10）、暂停（F9）、停止。“暂停”用于通知TBS暂停向TBC发包，此时由TBS缓存上下行数据。“停止”用于中断TBS向TBC发包，所有数据将直接按照TBS的规则进行传输。“继续”用于在暂停或停止状态下通知TBS恢复向TBC发包。

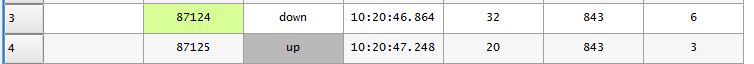
*TIPS：适当的时候选择暂停可以防止TBC列表中消息不断向下滚动。*

2.3.3、按钮组的前两个图标用于将当前选中的单个数据包进行上下移动，通过改变发送顺序，来模拟数据包乱序的场景。图标用于选择是否自动将数据列表的滚动条滚动到底部（可通过XML配置文件的中AutoScroll配置）。用于快速定位到最近一次选中的数据行上。图标仅在回放结束后的锁定状态下可用，用于帮助用户快速找到与当前录制DOWN数据行匹配的最近一次收到的数据行。

2.3.4、按钮组依次为“发送选中的数据”（F5）、“发送已标记的数据”（F6）、“标记为待发送数据”（F7）、“发送当前所有未发送数据”、“取消选中数据的发送标记”、“删除当前所有数据”。若某条数据被成功发送则其所在行的第一列将被标记为绿色，如：



若某条数据被标记为待发送数据，则其所在行的第二列将被标记为淡黄色（可配置其它颜色），如：



*TIPS：可通过双击某行数据的Seq列对其进行快速标记/取消标记。*

2.3.5、按钮组仅在加载录制数据时可用。其中用于播放已加载的录制数据，在播放过程中数据列表将被锁定，锁定后所有可以改变原有数据行位置的操作都会被禁用（例如：删除、插入、移动等）。用于在停止正在播放的数据。用于在播放结束后解锁数据列表。

*TIPS：数据录制与回放的内容详见下文*

* 1. 过滤工具栏



2.4.1、在该工具栏中，分别表示清空过滤规则和弹出过滤规则编辑框。用户可在或单击弹出的编辑框中编辑过滤条件（过滤条件的编写规范参见4过滤条件）。编辑完成后，单击或按回车键应用过滤条件，若应用成功则编辑框背景为绿色：

，若过滤过滤条件非法（语法错误）则编辑框背景红色：。编辑框背景为白色时表示当前过滤规则尚未应用。当有多个通道时，过滤条件将以通道为单位。不同通道的过滤条件之间不会相互影响。在切换通道时，过滤条件也将同步切换。

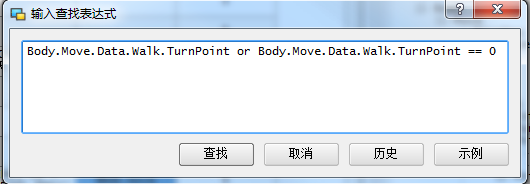
*TIPS：TBC能自动保存历史过滤条件，config.xml中RecentFilterNum配置项可以设置保存的最大条数。*

2.4.2、按钮用于在当前数据列表中查找符合查找表达式的数据。查找表达式使用类Python语法。以Wgame为例：

A）查找命令字大于10且BodyLen小于100的数据包：



B）查找含有Body.Move.Data.Walk.TurnPoint路径的数据包：



C）查找聊天内容为”xiyingshibada”的数据包：

Body.Message.Data.SysMsg.Content == "xiyingshiba"

D）查找Uin大于1000且命令字在{12, 23, 345}集合中的上行数据：

$Uin > 1000 and Head.Cmd in (12, 23, 345) and $\_DirectionName == “up”

**请注意以下事项：**

1、查找范围总是在当前所有数据包（包括隐藏数据包）

2、暂不支持节点值为中文的查找

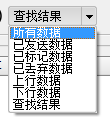
3、若要以某节点是否存在作为查找条件，最好使用类似：x.y.z or x.y.z == 0的结构，因为Python认为数值0为布尔假

4、查找完成后状态栏会显示一些简单信息，例如：

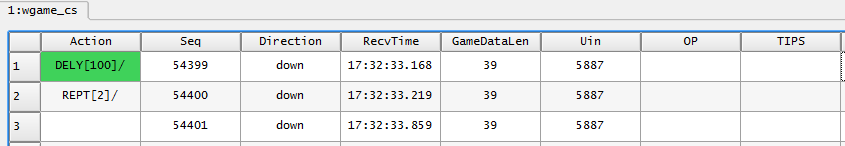


5、查找完成后，工具栏的数据类型选择框自动更新为“查找结果”，且其ToolTip会被设置为最近一次的查找条件：

若要重新查看所有数据包，则只需单击数据类型选择框，选择“所有数据”即可：



* 1. 数据列表视图



数据列表占据了大部分TBC界面，它以单元格的形式展示了所有数据的基本信息。默认情况下数据列表固定显示前6列数据，用户可通过config.xml配置文件在该列表中添加更多列（本示例添加了Cmd列，详见FQA部分）。该列表支持如下右键菜单：



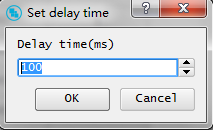
2.5.1、插入菜单允许用户在当前选中数据行的下一行以XML格式数据插入数据包。TBC将在检查XML数据格式通过后对使用XML数据跟节点指示的MetaName对插入XML数据进行打包。

2.5.2、复制菜单允许用户将当前选中的数据包复制到TBC内实现的剪贴板中，然后通过粘贴将已复制的数据拷贝到选中位置的上方。

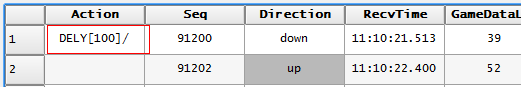
2.5.3、删除菜单用于删除已选中的数据（注意：该删除操作只发生在本地，TBC不会通知TBS该删除操作。为保证数据有序，不推荐用户在发送某个数据前执行对数据的删除操作。参见下文丢弃操作的介绍）。

2.5.4、删除自定义列菜单用于删除用户通过config.xml中的CustomNode配置项设置的或运行时添加的自定义列（如本示例中的Cmd列）

2.5.5、在附加操作中，TBC提供了延时、重复发包、TCP分包（暂不支持）、TCP粘包（暂不支持）和丢弃操作。用户的这些操作将通过TBCS之间的私有协议通知TBS完成。例如用户对某个数据包执行延时操作，在弹出的延时时间输入框中输入延时时间：

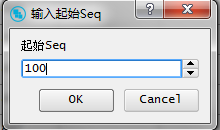


确定后该延时时间将显示在Action列：



TBC将该数据发送给TBS，由TBS对该数据执行100毫秒的延时。

2.5.6、自动生成序列号菜单用于批量修改数据包的Seq，用户只需选中需要更新Seq的数据行，单击右键-自动生成序列号菜单，然后在弹出的起始Seq对话框中设定起始Seq后单击OK即可：

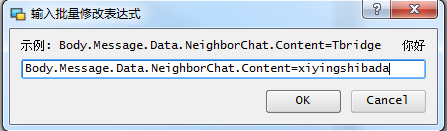


*TIPS：录制数据的Seq必须唯一。所以当需要录制的数据比较多，而你又不确定是否所有数据包的Seq都不相同时，可以试试这个功能。*

2.5.7、生成透传表达式菜单用于帮助用户快速可以透传选定数据包的过滤表达式。常用于如下场景：用户在没有对游戏客户端进行任何操作时，Tbridge收到的往往是心跳或安全验证相关的数据包，而这些数据对于用户的协议测试毫无用处，用户希望按命令字透传这些数据包。

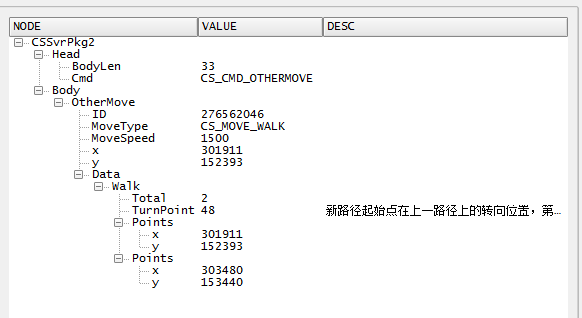
*TIPS：要使用该功能需要用户在config.xml中配置通道的命令字路径Cmd.Entry*

2.5.8、批量修改节点菜单用于按节点路径批量修改选中数据包的节点数据。以Wgame为例，需要修改批量修改聊天数据包的聊天内容字段时，只需选中数据包，右键选择批量修改节点菜单（或Ctrl+W）在弹出的对话框中按提示输入修改表达式即可：



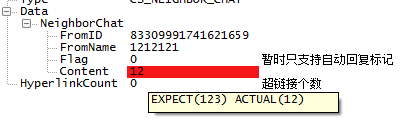
*TIPS：若选中数据没有修改表达式中的节点路径，则跳过。另外修改表达式右值左侧的空白符将被忽略。*

* 1. 协议树视图

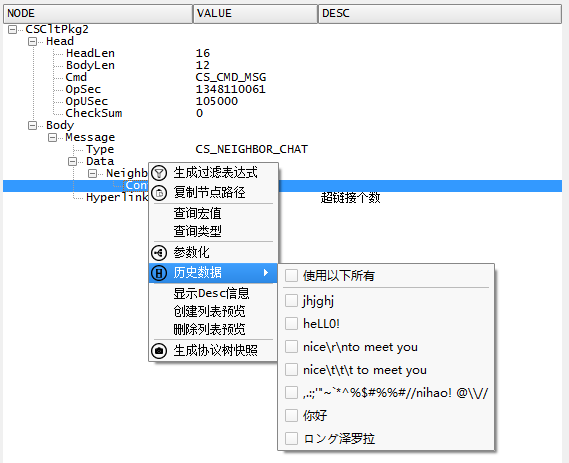


单击数据列表中的某行时，其游戏数据将在该协议树控件中显示。用户可双击VALUE列的数据进行编辑。DESC列的描述信息来自TBC加载的tdr协议文件，由TBC在创建协议树时自动生成。

在播放录制数据时，若收到的数据包与期望数据不同，TBC会在该视图分别使用红色和绿色背景标记期望失败和成功的字段，并在鼠标悬停于字段时显示期望与实际数据，例如：



协议树视图支持的右键菜单如下图所示：



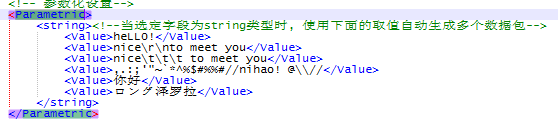
各菜单的介绍如下：

2.6.1、生成过滤表达式菜单使用选中节点的路径与数值生成用于过滤掉当前数据包的过滤表达式（参见4.过滤条件）。

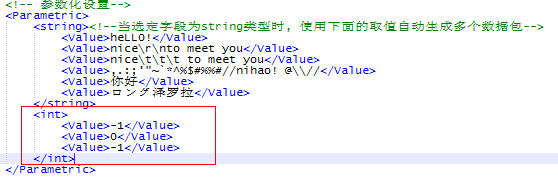
2.6.2、复制节点路径菜单用户将选定节点的路径复制到剪贴板，帮助用户快速书写OP指令（参见5.3.OP指令）。

*TIPS：双击节点也有同样的效果*

2.6.3、参数化菜单根据选定节点的数据类型将当前数据包扩展为N个。扩展数据包使用的数据优先以用户在config.xml文件中的Parametric配置为准，例如：

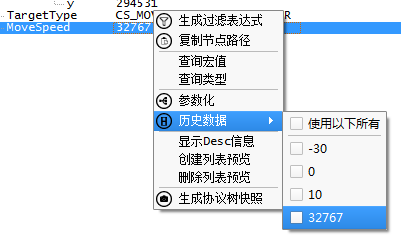


当需要在一个int类型的节点上使用参数化功能并使用数据-1，0，1时，可以扩展以上的配置为：



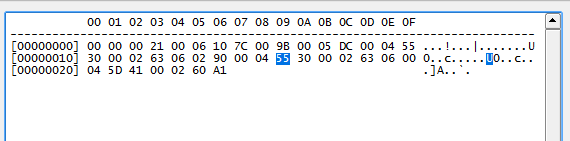
若选型节点的类型取值未在config.xml配置，则使用Tbridge内的类型边界值进行数据包扩展。

2.6.4、历史数据菜单提供了快速使用历史数据修改节点的方法。用户对协议树中节点的每次修改都将被记录到本地文件history.tbc中。这样在下次测试时，用户只需要之前修改过的节点上使用历史数据菜单，即可快速修改节点数据。如图：



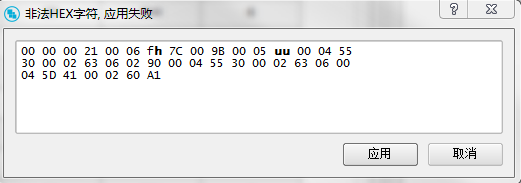
*TIPS：当数据包层次过深时，TBC界面可能会因为过深的递归计算出现假死现象。遇到该情形时，亲们只需耐心等待（大概5秒左右），可通过修改config.xml中的TreeExpandLevel配置项来缩短展开时间）。*

* 1. 二进制数据视图



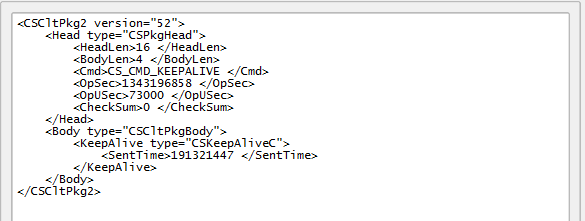
该视图显示选中行的二进制数据。双击某个字节时，TBC将自动查找该字节对应的ASCII字符，并将其高亮显示。

需要编辑时，单击右键——以文本模式编辑，即可在弹出的编辑框进行编辑操作。TBC会对修改后的数据进行合法性校验，当出现非HEX字符或字节数不是偶数时，TBC将提示数据错误，如图：



需要注意的是，当用户修改后的二进制数据无法通过TDR打包时，TBC会认为数据非法，并自动将数据回退到最近一次可正常打包时的数据。

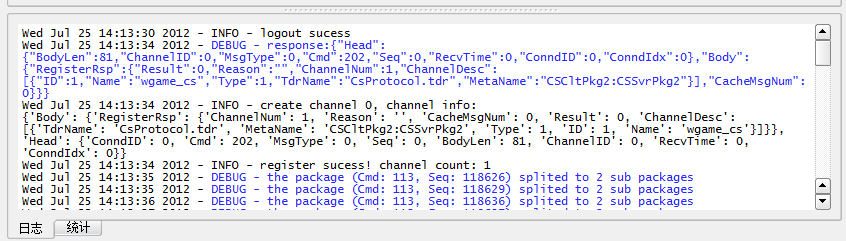
* 1. XML数据视图



该视图显示选中行的XML数据。用户可直接编辑该视图中的XML数据，编辑完成后使用右键——打包，对修改后的数据进行重新打包。若打包失败则弹出错误提示。

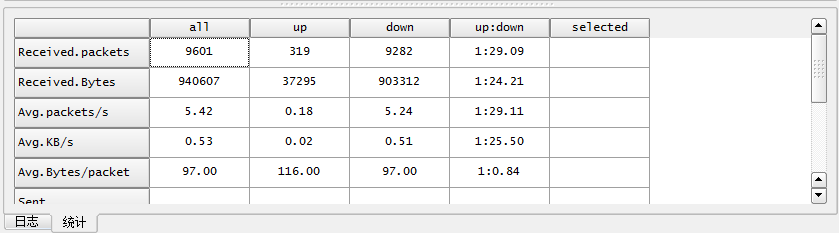
除打包菜单外，该视图右键还支持格式检查、导出和导入操作。分别用于检查XML格式、将XML导出到本地文件和从本地文件加载XML数据。

* 1. 日志



日志除在以上界面显示外，还将同步写入TBC所有目录的log/tbridge.log中。日志级别共有5种：CRITICAL、ERROR、WARNING、INFO、DEBUG。用户可在config.xml的LogLevel配置项设置或在右键弹出菜单中动态修改。

* 1. 统计



统计数据以通道为单位。切换通道时，通道数据也同步刷新。selected用于显示选中行的统计情况（暂未实现）。平均值默认每隔1秒计算一次（可在config.xml的StatInterval配置项设置）。

*TIPS：按住左侧的垂直标题栏，可上下拖动改变各统计项的显示顺序。*

* 1. 状态栏



TBC将状态栏划分为5列，依次用于显示运行时消息、状态、选中行列、简要统计和耗时信息。

配置文件

TBC配置使用XML文件格式，默认为config.xml。由通用配置项目和通道相关配置项构成。用户可在发布包的根目录下找到该配置文件。具体配置项含义请参考XML文件中的注释。示例如下：

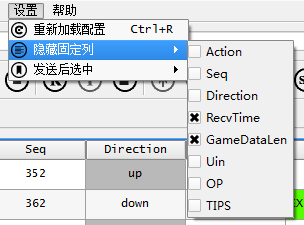


* 1. 关键配置项

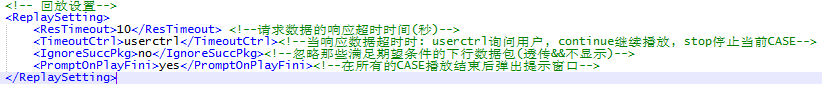
3.1.1、<HideColumn>用于配制需要隐藏的固定列。固定列表示与具体游戏业务无关的 数据列：Action、Direction、GameDataLen、Seq、RecvTime、Uin、OP、TIPS。如果用户不希望在当前的测试中显示其中的某些列，则可以在该配置中将其隐藏，例如：



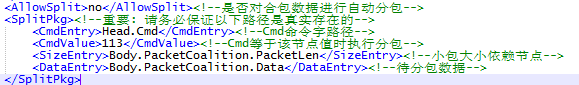
此外用户也可在启动TBC后通过配置-隐藏固定列菜单动态隐藏/显示：



3.1.2、< ReplaySetting>用于配置播放录制数据时的行为。当前支持的配置与含义如下图所示：

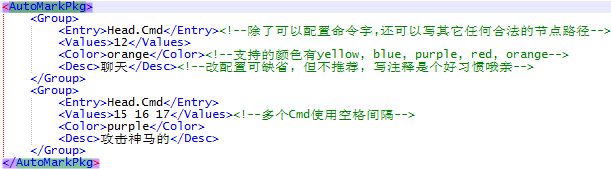


3.1.3、< AllowSplit>与< SplitPkg>用户配置TBC的拆包策略。之所以需要拆包，是因为公司的一些游戏（如Wgame）常常会将多个数据包合并在一个大包中进行传输，如果TBC不对这些大包进行二次拆包，我们就不能看到隐含子包数据结构。一个针对Wgame的示例配置如下：

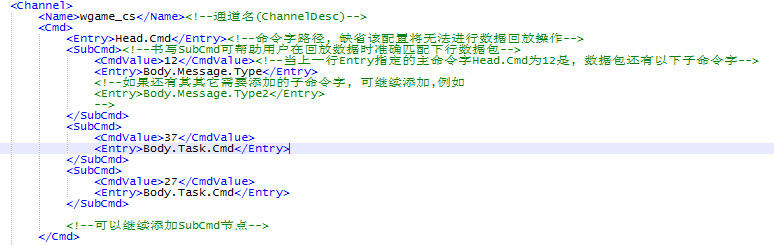


*TIPS：基于效率上的考虑TBC并未对以上配置的合法性做严谨判断，因此请务必在熟悉游戏的合包结构后再认真配置。*

3.1.4、< AutoMarkPkg>用于将满足条件的数据包自动标记为待发送数据。一个针对Wgame的示例配置如下：



3.1.5、<Cmd>用户配置数据包的命令字路径。如图：



Cmd.Entry标识主命令字的路径（只可以有一个主命令）。SubCmd表示子命令字，可以配置多个。SubCmd.CmdValue标识主命令的值，SubCmd.Entry表示对应主命令字值的子命令字路径。例如上图所示，在Wgame游戏中，主命令字为12时的聊天数据包有子命令字Body.Message.Type。

*TIPS：TBC在回放数据等操作时依赖Cmd.Entry，因此若要保证一切可顺利执行，就一定将该项配置妥当。*

过滤条件

过滤条件分为普通模式的过滤条件和高级模式的过滤条件。除了需要使用花括号包裹节点路径表示对某个节点的引用外，其它语法均与Python一致。

1. 普通模式的过滤条件如下：



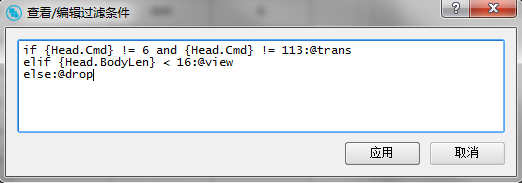
该模式下，用户只需使用一对花括号包裹某个Entry节点的全路径，再对其进行布尔运算即可。如果数据包满足过滤条件则显示在数据列表中，否则将被透传回TBS。

除and表示与操作外，普通模式还支持not和or操作。例如：



*TIPS：在协议树的Entry节点上单击右键->生成过滤表达式，可快速创建普通模式的过滤表达式。*

2）高级模式的过滤条件如下：



该模式以if关键字开头，其使用的布尔表达式规则与普通模式一致。此外，该模式支持自定义的@trans、@view、@drop和@delete动作。其中@trans表示透传符合条件的数据包，@view表示透传并界面显示符合条件的数据包，@drop表示通过服务器丢弃符合条件的数据包，@delete表示本地丢弃符合条件的数据包。

若要按固定列字段进行过滤，则需要在{}前加英式$符号。当前支持过滤的固定列字段有：${Seq}、${Direction}（取值为”up”/”down”）、${ConndIdx}。示例：

1、透传并显示所有数据包：



2、透传所有命令字不在[3, 6, 9,12]中的数据包：



3、通知服务器丢弃包体大小在[1,100]的数据包：



4、透传所有上行数据且透传并显示所有下行数据：



*TIPS：在你的测试中，也许命令字或包体大小并不是Head.Cmd和Head.BodyLen*

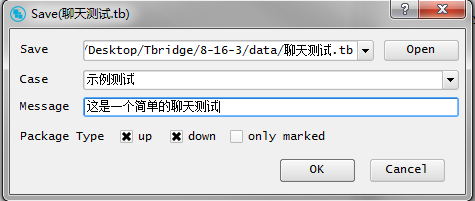
*如果需要使用if…else…模式的过滤表达式，则必须单击按钮，在弹出的过滤表达式编辑框中进行编辑（因为Python要求严格的代码缩进格式，单行编辑可能会导致语法错误哦亲！）*

录制与回放

从1.0.2版本开始，TBC支持数据的录制与回放操作。录制与回放致力于用于解决协议测试中多次重复测试同一场景的重复性劳动问题。

* 2. 如何录制数据

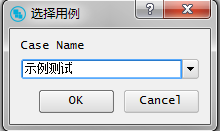
用户可在连接后的任意时刻执行录制（保存数据）操作。单击开始-保存菜单或使用Ctrl+S快捷键即可弹出保存数据对话框：



录制数据时，一个文件被视作一个TestSuite，用户可在同一文件中保存多个TestCase。可通过up、down、onlymarked复选框指定保存数据的类型。默认up和down会被选中，即保存上行和下行数据。当勾选onlymarked时，默认up和down被自动勾选，表示仅保存当前已标记为待发送状态的数据。

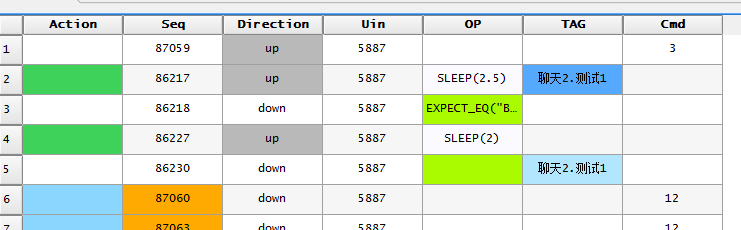
* 1. 如何回放数据

在回放数据前，用户需要加载已经录制的数据。单击开始-打开菜单或使用Ctrl+O快捷键即可弹出打开文件对话框，选中录制文件（以tb为扩展名）后。若文件可被成功加载则弹出用例选择对话框，如图：



TBC会在该对话框中列出存在于打开文件的所有可用Case，用户选定Case后单击OK按钮即可完成录制数据的加载。

加载数据完成后单击工具栏按钮组中的按钮启动回放，如图：

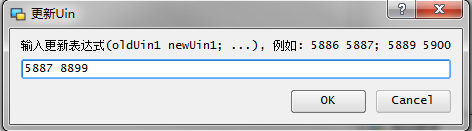


如上图所示，TBC在回放数据时总是在收到请求数据的所有响应或响应超时后才会继续播放下一条请求数据。默认的响应超时为3秒，用户可在任意需要重置超时时间的UP数据包OP列使用TIMEOUT(SEC)设置超时时间，例如：TIMEOUT(2)

需要注意的是回放数据时必须保证录制数据时的游戏客户端已登陆且回放数据的Uin必须与登陆客户端的Uin一致。

*TIPS：若用户确定Uin不一致对其测试无影响，则可不比保证Uin的一致性。*

当需要修改加载数据的Uin时，可在CASE首行的Uin列处双击，在弹出的修改Uin对话框中根据提示进行修改：



如上图所示，用户单击确定后，Tbridge将替换CASE中所有5887的Uin为8899。

* 1. OP指令

如上图所示，用户可为数据添加回放控制指令（OP指令）。添加的OP指令会在录制数据时与业务数据一起保存到本地文件。在数据回放时，TBC解析并执行OP指令完成协议测试自动化。

当前支持的OP指令有：

**MOV**：将数据包的某个字段值保存到TBC全局寄存器中

MOV(“R0”, 1024)

MOV(“R1”, “Body.Move.MapIdx”)

**SLEEP**：休眠N秒

SLEEP(2.5)

**TIMEOUT**：设置请求数据的响应超时，默认为3秒

TIMEOUT(1.5)

**UPDATE:** 更新字段数据

UPDATE(“Body.Move.MoveSpeed”, 5500)

UPDATE(“Body.Message.Data.NeighborChat.Content”, “你好world”)

MOV(“R1”, “你好”)

UPDATE(“Body.Message.Data.NeighborChat.Content”, “R1”)

**INC：**创建一个自增计数器

INC(“C0”, 1) #创建一个初始值为1的自增计数器，初始化操作尽在第一次执行该命令时有效

**DEC：**创建一个自减计数器

DEC(“C1”, 10) #创建一个初始值为10的自减计数器，初始化操作尽在第一次执行该命令时有效

**JMP：**跳转到Seq行

JMP(1024)

**JLP：**循环跳转N次

JLP(3, 1024) #三次跳转到Seq为1024的数据行

**JNZ：**当表达式非零时执行跳转

DEC(“C1”, “3”)

JNZ(“C1”, 1024) #与JLP(3, 1024)等效

**JIF：**当条件为真时跳转到Seq1否则跳转到Seq2

JIF(“Head.Cmd == 12 and Head.BodyLen > 90”, 1024, 19) #当布尔表达式为真时跳转到Seq为1024的数据行，为假时跳转到Seq为19的数据行

JIF(“Body.Message.Data.NeighborChat.Content != ‘hello’”, 230) #当布尔表达式为真时跳转到Seq为230的数据行，为假时不做任何跳转

**CALL：**将已经加载的CASE视作一个可被调用的函数，通过CALL指令呼叫执行

假设已经加载CASE：聊天2.测试1



则可在任意CASE的UP数据包中通过CALL(“聊天2.测试1”)指令调用之。被调用的CASE将从第一个UP数据开始执行，执行结束后返回到原来调用指令所在UP的下一个UP数据包处。

**EXPR：**计算数学表达式的值，并将其存储到全局寄存器中

EXPR(“R1”, “1023 + 1 / 5.5 \* 120”)

EXPR(“R2”, “R1 \* 10”)

EXPR(“R3”, “Head.BodyLen / 2”)

注意：EXPR仅用于数学运算，不要在表达式中使用中文

**NOP：**空操作指令

**NOP()**

注意：NOP指令仅可在UP数据包中使用，TBC将不发送使用该指令的UP数据包

**EXPECT\_EQ**：期望数据包的某个字段与字符串相同。

EXPECT\_EQ(“R0”, 1024)

EXPECT\_EQ(“Body.Move.MoveSpeed”, 4400)

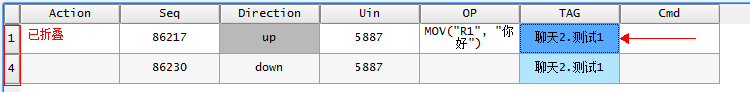
*TIPS：EXPECT系列还有函数EXPECT\_NE，EXPECT\_LE，EXPECT\_LT，EXPECT\_GE，EXPECT\_GT。其函数与GoogleTest中定义一致，用法与EXPECT\_EQ相同。*

* 1. TAG列

TAG列不仅用于数据的录制回放，用户可以在平时的协议测试中在任意数据行双击该列队数据添加备注信息。

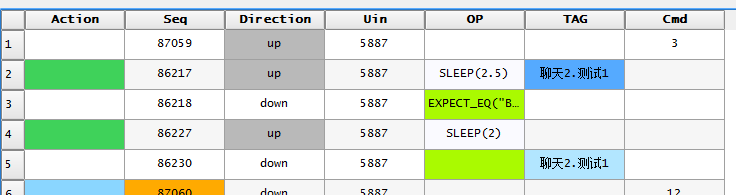
TAG信息会在录制数据时一同保存到本地文件。下次加载时又会重新出现，因此对重要数据包数据TIPS备注是个不错的习惯。

特殊的，当在加载CASE的首行双击TAG列时，TBC不会弹出TAG编辑框，而是将当前CASE折叠/展开。如图：



* 1. UP与DOWN的对应关系

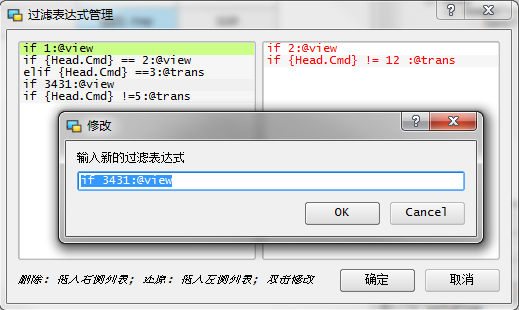
也许你会问回放数据时TBC怎么知道一个请求的响应数据有哪些呢？对此，TBC遵循“所见即所得”的原则，即TBC将当前UP会下一个UP数据包之间的所有DOWN数据包作为当前UP的DOWN数据包。这与界面中所展示的数据关系是对应的。因此，对于一个简单的聊天测试，我们在录制数据时需要将每个聊天请求和它对应的聊天响应依次对应隔开：



管理面板

* 1. 过滤表达式管理面板

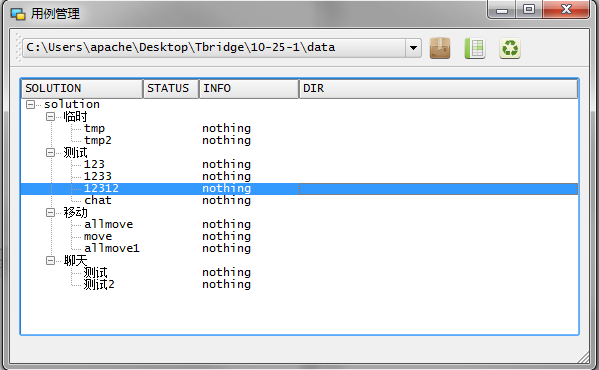
单击按钮弹出过滤表达式管理面板，用户可在该面板中修改/删除已有过滤条件。如图：



双击左侧列表中的过滤表达式弹出修改对话框。

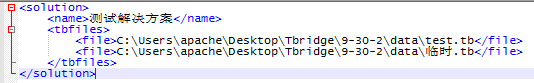
* 1. 用例管理面板

单击开始—用例管理或快捷键Ctrl+M弹出用例管理面板：



按钮依次表示：打开文件夹、打开解决方案、更新。

用例展示使用如上图所示的Solution.SuiteName.CaseName的树形结构。打开文件夹按钮用于遍历目录下的所有Suite（\*.tb文件）并以默认的解决方案名”solution”作为根节点。打开解决方案按钮用于打开扩展名为tsln的解决方案描述文件，该文件定义了解决方案名称与需要打开的Suite文件路径，例如：

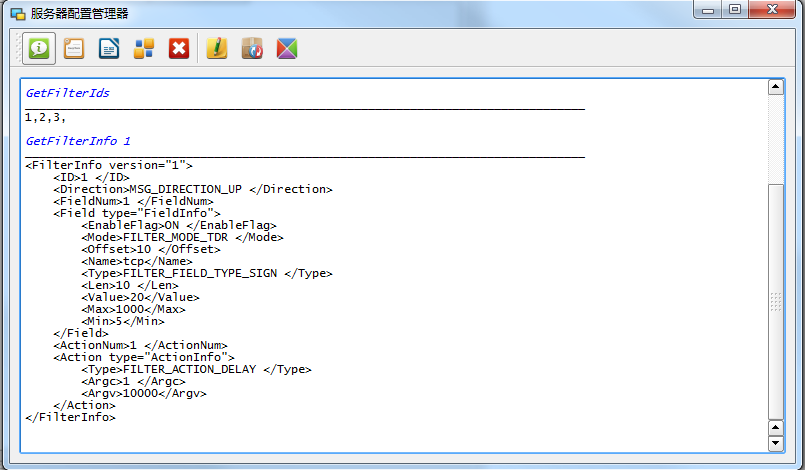


当已打开的目录或解决方案变更（删除或增加一个suite等）时，用例管理面板中的数据不会被立即同步，需要用户手动单击按钮进行数据更新。

要加载用例时，只需将对应节点拖入数据列表中即可。

* 1. 服务器配置管理面板

该面板用与在客户端管理TBS部分配置。单击开始—服务器配置管理或快捷键Ctrl+N弹出服务器配置管理面板：

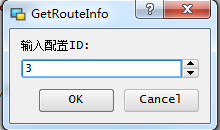




以上工具栏按钮依次为：

获取过滤配置ID列表、获取过滤配置（需要配置ID）、获取路由配置（需要配置ID）、删除过滤配置（需要配置ID）、添加过滤配置（需要配置XML）、更新过滤配置（需要配置XML）、更新路由配置（需要配置ID和XML）。

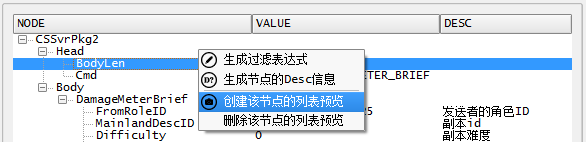
例如要获取ID为3的路由配置，则单击按钮，在弹出的ID输入框中输入3，单击OK即可：



FAQ

* 1. 如何在数据列表中添加协议树中某个Entry节点的预览？

A：在Entry节点上右键，选择创建该节点的列表预览



或者直接将节点拖拽到左侧的数据列表。

* 1. 如何清空已经添加到数据包的Action操作？

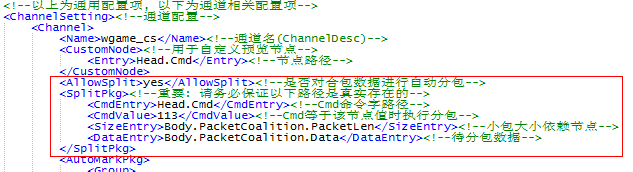
A：直接在对应的Action单元格上双击即可。

* 1. 能不能在上下行数据包使用的MetaName一致时切换数据包的方向？

A：虽不推荐这么做，但还是可以通过双击Direction列来切换方向（仅在上下行数据包使用相同的MetaName时可用）。

* 1. 我的游戏协议中有很多合包，有木有办法自动拆开合包呢？

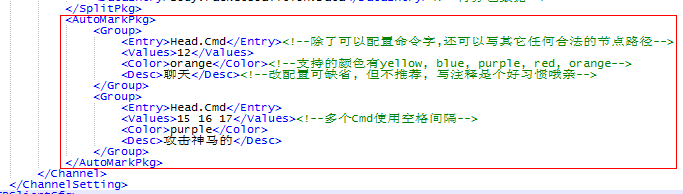
A：可以的。只需在TBC的XML配置中写好拆包参数即可。以Wgame为例，我们的自动拆包规则如下：



注意：为提高拆包效率，当前TBC未对拆包参数进行合法性校验，所以配置这个参数时务必填写正确的Entry路径。

* 1. 如何自动将满足条件的数据包标记为待发送数据？

A：使用TBC的AutoMarkPkg配置项可实现这个功能。以Wgame为例，我们希望在收到12号命令字的数据时将其标记为待发送数据（使用橙色标记），在收到15、16和17好命令字的数据时也将其标记为待发送数据（使用紫色）。那么我们可以这样配置客户端：



* 1. 登陆时TBC提示找不到tdr文件怎么办？



A：只需将tdr文件放到lib目录下即可（lib与TBC同目录）。但需要注意，不可将Linux下的tdr文件直接拿到Windows下使用。你需要使用tools目录下的tdr.exe重新生成Windows下使用的tdr文件。例如：



* 1. 我确定TBS、游戏客户端和服务器都正常，但TBC就是没反应怎么办？

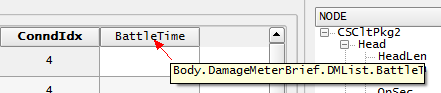
A：深表遗憾，这是很可能TBC已经出现了未知异常停止工作了。建议你关闭客户端，留意关闭后是否有error日志提示，若有请将其发送给Apache进行错误定位。

* 1. 为什么有些快捷方式不能用？

A：额，这个我们会在下个版本支持，敬请期待！

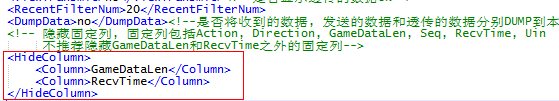
* 1. 如何查看自定义列的全路径？

A：鼠标悬停在自定义列上，稍等即见。



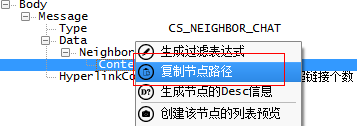
* 1. 可不可以隐藏一些我不关注的固定列，比如GameDataLen？

A：可以，在XML配置中使用如下格式隐藏固定列



* 1. 我在编写OP指令时经常要手动编写期望字段的路径，有木有简单一点的办法？

A：好问题！试试在某个字段上右键吧！



* 1. 我希望TBC能在我F5发送当前行数据后，立即选中下一行数据，这个支持么？

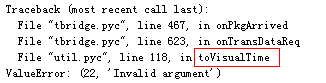
A：我们也注意到了这个问题，并提供了XML配置项和相应的菜单来实现你的愿望。你不仅可以在XML配置中使用进行静态配置，还可以在TBC启动后单击**设置-发送后选中-下一行**菜单实现需求。

* 1. 我修改了一些XML配置，但又不想重启TBC，可以Reload配置么？

A：看图不解释。



* 1. 出现了如下的异常时怎么办？



A：请将config.xml中的UseGWT配置为yes后再试。