SEQUENCER介绍

文档历史

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 作者 | 描述说明 |
| 2016-3-26 | 1.0 | 李晗 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 文档最后一次修改时间 | | 2016-03-27 15:47:00 | |

目 录

1 SEQUENCER简介 1

1.1 标题二 1

1.1.1 标题三 1

1.1.1.1 标题四 1

1.1.1.1.1 标题五 1

1.1.1.1.1.1 标题六 1

2 详细说明 3

3 使用方法（基于Office Word 2003） 6

3.1 模板与样式的定义 6

3.2 模板与样式使用方法 6

3.3 模板的保存 6

4 操作流程 8

5 文档约定 9

5.1 前言 9

5.2 读者对象 9

5.3 通用格式约定 9

5.4 图形界面格式约定 9

5.5 键盘操作约定 9

5.6 鼠标操作约定 10

5.7 各类标志 10

5.8 其他 10

6 注意事项 11

7 附件 12

7.1 如何修改、增加样式： 12

7.2 标题多级编号 12

7.3 Word模板使用全解析 12

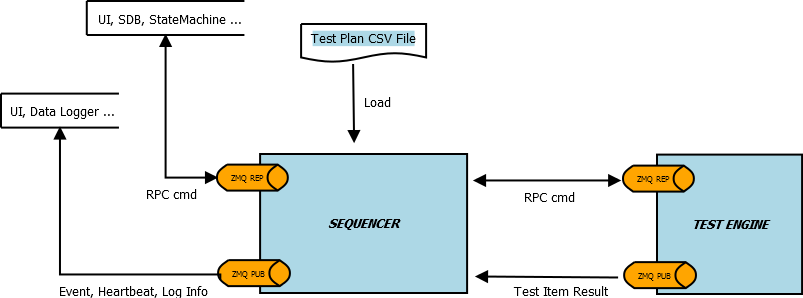
7.4 Word长篇文档排版技巧.doc 12

# SEQUENCER简介

## 基本功能职责

SEQUENCER组件的基本职责是基于预定义的测试计划，通过控制TestEngine执行测试来完成全部测试流程，并对每一项测试任务的执行结果以及整体测试计划的执行结果进行判断，然后通过ZMQ PUB端口将测试计划的实时执行情况进行发布。

## 接口及交互



如图所示，SEQUENCER组件的输入信息包括csv格式的测试计划定义文件以及通过ZMQ RPC端口传入的控制指令；输出信息基于ZMQ PUB端口进行发布；同时基于ZMQ RPC及PUB通讯控制TEST ENGINE执行具体的测试任务。

## SEQUENCER启动及端口说明

SEQUENCER的启动命令为

python sequencer/sequencer.py -s [index]

因为SEQUENCER可以启动多个实例，启动指令中最后一个数字就是某个SEQUNCER实例的索引标识，从0开始。

SEQUENCER组件的ZMQ PUB及RPC端口分别由配置文件zmqports.json中的"SEQUENCER\_PUB"和"SEQUENCER\_PORT"项的数值决定，每个启动实例监听的实际端口号为配置端口+实例索引index。

* 图片格式说明：

图片版式设置为上下型加居中。

图 1.1 XXXXXXX（已设置好域，复制使用即可）



重点、要点或总结：重点、要点或

# 测试计划（Test Sequence）

测试计划定义是CSV文件，文件名格式为[name]\_[version].csv。其中[name]为 测试计划的名称，[version]则为版本号。名称和版本信息会在用户界面中呈现。Name和Version中间的分隔符为“\_\_”-两个“\_"，因此，Name和Version中可以包含单个的”\_“符号，但不能包含双"\_"。

测试计划文件名示例：Han\_SimpleTest\_\_0322\_11h.csv

测试计划定义文件的具体格式详见“TestSequenceSpec.pdf”文档。

# ZMQ PUB端口信息

## PUB端口信息基本格式

Sequencer的信息通过ZMQ的pub端口发布出去。基本的消息格式由五个部分组成，依次为：topic，timestamp，verbosity level，publisher identity，data，各个部分的说明如下：

|  |  |
| --- | --- |
| PUB端口信息格式字段说明 | |
| 字段 | 说明 |
| topic | PUB消息的主题，subscriber可以基于该字段对数据进行过滤；Sequencer的所有pub消息topic统一为”101” |
| timestamp | 调用publish方法发布消息的时间点，格式为 hh:mm:ss.ms |
| verbosity level | 指示当前的消息级别，目前的定义为：REPORTER=0，CRITICAL=1,INFO=2,DEBUG=3 |
| publisher identity | 表明当前的消息发布者，SEQUENCER PUB消息对应字段为”sequencer\_[index]” |
| data | 实际的消息数据，不同类型的消息对有不同的格式定义和规定。 |

表 1.1 PUB端口信息格式字段说明

接收PUB端口信息的Lua代码示例：

topic, timestamp, level, publisher, data = s:recv\_multipart()

print("received:", topic, timestamp, level, publisher, data)

## PUB端口通道的信息分类

SEQUENCER的PUB端口通道包含三类信息：

### Event消息

报告测试进度及结果的事件信息，对应消息的verbosity level为0，data消息体部分为json字符串。格式为{event:[event];data:[data]}。

|  |  |
| --- | --- |
| Event信息基础字段说明 | |
| event | 事件类型，取值为以下之一 0=SEQUENCE\_START, 1=SEQUENCE\_END, 2=ITEM\_START, 3=ITEM\_END, 4=ATTRIBUTE\_FOUND |
| data | 事件的扩展信息，内容为json字符串，不同event类型的对应的字段定义见下表 |

|  |  |  |  |
| --- | --- | --- | --- |
| 各类Event信息字段定义说明 | | | |
| SEQUENCE\_START | **json格式** | | **说明** |
| {name:[sequence name]; version:[sequence version]} | | 测试计划的name和version信息 |
| **示例** | {"data": {"version": "0322\_11h\_1.0.0", "name": "hantest\_plan"}, "event": 0} | |
| SEQUENCE\_END | **json格式** | | **说明** |
| {result:[result];error:[error message];logs:[logpath]} | | result表示sequence执行结果，0:fail,1:pass,-1:aborted；error字段只有在result=-1时出现。logs字段包括相关的日志文件路径。 |
| **示例** | {"data": {"result": false, "logs": "."}, "event": 1} | |
| ITEM\_START | **json格式** | | **说明** |
| {"group": [group], "to\_pdca":, "high": , "low":, "tid":, "unit":} | | group：所属分组；tid：测试项id；unit：结果单位；low，high：结果阈值范围；pdca：是否需要将结果发送到PDCA，True/False |
| **示例** | {"data": {"group": "HANTESTGROUP", "to\_pdca": true, "high": "50", "low": "5", "tid": "INTEL\_HOG\_100\_STAT\_somefail", "unit": ""}, "event": 2} | |
| ITEM\_END | **json格式** | | **说明** |
| {"tid":,"to\_pdca":,"result":, "value":, error” } | | tid: 测试项id; value：测试结果；result：测试项执行结果，取值为：0:fail,1:pass,-1:aborted；error:错误信息，只有在result=-1时出现 |
| **示例** | {"data": {"tid": "INTEL\_HOG\_100\_STAT\_somefail","to\_pdca": true, "result": false, "value": 123}, "event": 3} | |
| ATTRIBUTE\_FOUND | **json格式** | | **说明** |
| {“name”:, “value”: } | |  |
| **示例** |  | |

。

### Heartbeat消息

SEQUENCER心跳包信息，默认发送周期为5s。verbosity lever为3，data消息体为” FCT\_HEARTBEAT”。

消息示例：

“101 03-25\_15:36:41.291000 3 Sequencer\_0 FCT\_HEARTBEAT”

### Log日志消息

SEQUENCER的日志消息，记录SEQUENCER接收到的RPC指令消息。verbosity lever为3，data消息体为日志内容。

消息示例：

“101 03-25\_15:43:55.048000 3 Sequencer\_0 received msg {"function": "run", "params": null, "jsonrpc": "1.0", "id": "53714280f25d11e59f58acbc32d422c0"}”

# RPC端口指令

## RPC端口控制指令基础协议

SEQUENCER通过ZMQ Request-Reply模式的服务端口接收RPC控制指令，RPC控制指令的请求协议格式为Json字符串，包含以下字段：

|  |  |
| --- | --- |
| RPC端口请求信息格式字段说明 | |
| 字段 | 说明 |
| jsonrpc | 协议版本，当前为“1.0” |
| id | 当前请求的唯一id |
| function | 请求命令类型，取值为：status/run/load/show/jump/list/next/step/abort/wait |
| params | 执行命令所需要的参数 |

响应消息的格式同样Json字符串，包含以下字段：

|  |  |
| --- | --- |
| RPC端口响应信息格式字段说明 | |
| 字段 | 说明 |
| jsonrpc | 协议版本，当前为“1.0” |
| id | 当前请求的唯一id |
| result | 请求指令的执行后的返回值 |
| error | 如果执行产生异常或错误，会在此包含错误信息 |

各指令对应的params参数详细说明参见“SequencerRPC.pdf”文档。

消息示例：

请求消息：

{"function": "load", "params": ["/Hantest\_plan\_\_0322\_11h.csv"], "jsonrpc": "1.0", "id": "cmd1"}

响应消息：

{"jsonrpc": "1.0", "id": "cmd1", "result": "/Hantest\_plan\_\_0322\_11h.csv has been loaded"}

## sdb介绍

SDB是一个基于命令行的Sequencer调试工具，除了以上的基本指令外，sdb还额外支持4个用于调试的指令，介绍如下

|  |  |
| --- | --- |
| sdb调试指令 | |
| 指令 | 说明 |
| continue | 协从当前的测试任务开始执行到下一个断点位置。（这种方式Sequencer并不支持，是由sdb通过单步操作来模拟实现的） |
| break | 测试计划行号；设置断点 |
| all | 无，显示所有断点位置 |
| timeout | 超时时长（ms） 修改对Sequencer指令请求的超时时长，为0时表示无超时时长 |

下面是一个sdb操作流程示例：

