[1. 承载与连接 3](#_Toc397607953)

[1.1 概述 3](#_Toc397607954)

[1.2 分层 3](#_Toc397607955)

[1.3 连接建立与释放 3](#_Toc397607956)

[1.4 连接异常 4](#_Toc397607957)

[1.5 消息分隔 4](#_Toc397607958)

[2. RF与PSTT接口（Command） 5](#_Toc397607959)

[2.1 接口格式 5](#_Toc397607960)

[2.2 SendScript 5](#_Toc397607961)

[2.3 ScriptRecieved 5](#_Toc397607962)

[2.4 Start 5](#_Toc397607963)

[2.5 Started 6](#_Toc397607964)

[2.6 Event 6](#_Toc397607965)

[2.7 Variable 7](#_Toc397607966)

[2.8 ScriptComplete 7](#_Toc397607967)

[2.9 Stop 8](#_Toc397607968)

[2.10 Stopped 8](#_Toc397607969)

[2.11 Report 8](#_Toc397607970)

[2.12 Exception 9](#_Toc397607971)

[3. PSTT测试脚本（Script） 9](#_Toc397607972)

[3.1 脚本格式 9](#_Toc397607973)

[3.2 SendMessage 9](#_Toc397607974)

[3.3 ReceiveMessage 10](#_Toc397607975)

[3.4 SendEvent 11](#_Toc397607976)

[3.5 ReceiveEvent 11](#_Toc397607977)

[3.6 Wait 12](#_Toc397607978)

[3.7 Finish 12](#_Toc397607979)

[3.8 RetrieveData 12](#_Toc397607980)

[3.9 SetUserGroup 14](#_Toc397607981)

[3.10 SetStratergy 14](#_Toc397607982)

[4. Retrieve Data中各个操作的参数 14](#_Toc397607983)

[4.1 MessageCount 14](#_Toc397607984)

[4.2 MessageParameterCount 15](#_Toc397607985)

[4.3 MessageParameterValue 15](#_Toc397607986)

[4.4 MessageOrder 15](#_Toc397607987)

[5. 关键字参数与测试脚本参数的映射 16](#_Toc397607988)

[5.1 Send 16](#_Toc397607989)

[5.2 OnReceive 16](#_Toc397607990)

[5.3 Event 16](#_Toc397607991)

[5.4 Wait 16](#_Toc397607992)

[5.5 Finish 17](#_Toc397607993)

[5.6 RetrieveData 17](#_Toc397607994)

[5.7 SetUserGroup 17](#_Toc397607995)

[5.8 SetStratergy 17](#_Toc397607996)

[6. 测试结果异常 18](#_Toc397607997)

[6.1 通信异常 18](#_Toc397607998)

[6.2 测试执行异常 18](#_Toc397607999)

# 承载与连接

## 概述

RF与PSTT间采用TCP方式作为承载，RF作为TCP客户端，测试工具作为服务端。在TCP之上，RF和PSTT使用JSON格式描述RF与PSTT间的消息。

RF执行每个测试用例时，将触发向PSTT建立连接，测试脚本执行完成后释放连接。

对于工具而言，连接是以进程为单位建立的，不需要为每个模拟网元单独建立一个连接。

如果在一个测试用例执行过程中通信异常，则该测试用例失败。由于每个测试用例独立建立连接，因此通信异常仅波及一个用例。

## 分层

在TCP之上，RF与测试工具间的协议分为两层，底层为Command层，是RF与所有测试工具间的通用接口，不管测试工具做何种用途，都需要通过这一层来与RF对接。

在Command上为Script层，这一层可以根据测试工具的特点，为该工具定制的业务原语，并在测试用例中编写使用。

Command层保证了各种不同的工具能够使用统一的接口与RF对接，Script层保证了各种不同用途的工具，能够根据自己的需求定制所需的操作。

Command和Script层都使用JSON来描述数据格式。



## 连接建立与释放

测试工具通过TCP三次握手建立连接。在应用层，不需要额外的过程再为模拟网元建立会话，RF与测试工具间的每条消息中都会携带模拟网元名称。

RF执行测试用例时，执行到Finish时触发向测试工具建立连接，在测试用例执行结束后（收到所有Report或者Exception），RF向测试工具拆链。

严格来讲，RF与测试工具间的连接并不与测试用例的执行对应，一个测试用例中，可能会有多次连接，如Setup、测试执行、Teardown分别对应一个连接。为了保证一个测试的连续性，测试工具中保存的变量、上下文等不应该随连接的释放来清理，而应该在测试用例中使用额外的参数来标识。因此，RF与测试工具间连接的建立与释放，不影响保存的上下文与数据区。



## 连接异常

RF和工具各自检测连接异常。

当RF发现连接异常后，将标记测试用例失败，并停止所有仍然可及的所有工具。

当工具发现连接异常后，将终止正在执行的测试脚本，并清理所有上下文相关内容，将测试工具恢复到初始状态。这里清理上下文的原因是，上一用例的执行已经不受控，因此上下文中的内容是非预期的。另外，连接中断也可能发生在RF的Teardown中，这时就没有其他途径去释放上下文了。

如果工具中，上一个用例还没有停止就收到了RF的SendScript，工具放弃上一用例的执行，清理所有上下文相关内容，准备开始执行当前测试用例。被放弃的上一个测试用例，不需要向RF上报信息。

## 消息分隔

TCP是面向字符流的协议，在TCP层并不提供消息分隔的方法，因此需要在应用层自行拆分数据流，从数据流中提取出一个个的消息。

本方案中，应用层采用JSON格式的字符流，为简化协议处理，在每个JSON描述前增加长度域，长度为4字节，按网络字节序传输，长度值为JSON描述的长度，不包括长度域自身的长度。

## JSON

在编码后的JSON描述中，除长度域外，不可以包含不可打印字符。

JSON描述中，大小写敏感。

# RF与PSTT接口（Command）

## 接口格式

RF与PSTT间使用JSON格式编码的数据。所有消息的基本格式如下：

**{**

**“type”: <message type>,**

**“office”: <simulator office name>,**

**<other command related parameters>**

**}**

## SendScript

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | SendScript | | |
| **用途** | RF向PSTT发送脚本 | | |
| **方向** | RF->PSTT | | |
| **参数** | **名称** | **类型** | **参数说明** |
| Script | 数组，复合类型 | 需要执行的脚本 |
| **编码举例** | {  "type": "SendScript",  "office": "enb1-mme1",  "script": [  <step>,  <step>,  …  ]  } | | |
| **备注** | 脚本格式参见第3节，脚本可以为空。 | | |

## ScriptRecieved

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | ScriptRecieved | | |
| **用途** | PSTT已经收到脚本时发送给PSTT | | |
| **方向** | PSTT->RF | | |
| **参数** | **名称** | **类型** | **参数说明** |
| - | - | - |
| **编码举例** | {  "type": "ScriptReceived",  "office": "enb1-mme1",  } | | |
| **备注** | 只能在工具收到SendScript后发送，不可独立使用 | | |

## Start

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Start | | |
| **用途** | RF要求PSTT中的模拟网元开始执行已经分发的脚本 | | |
| **方向** | RF->PSTT | | |
| **参数** | **名称** | **类型** | **参数说明** |
| timer | 数值 | 测试用例执行的最长持续时间(s)，0为无限 |
| **编码举例** | {  "type": "Start",  "office": "enb1-mme1",  “timer”: 10  } | | |
| **备注** | 只能在SendScript后执行  timer超时后，测试用例仍未停止，测试工具上报Exception | | |

## Started

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Started | | |
| **用途** | PSTT已经收到Start并开始执行脚本 | | |
| **方向** | PSTT->RF | | |
| **参数** | **名称** | **类型** | **参数说明** |
| - | - | - |
| **编码举例** | {  "type": " Started",  "office": "enb1-mme1",  } | | |
| **备注** | 只能在工具收到Start后发送，不可独立使用。  不需要等待脚本执行完毕，脚本一旦开始执行即可发送该消息。 | | |

## Event

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Event | | |
| **用途1** | PSTT向RF上报事件 | | |
| **用途2** | RF将收到的事件广播给所有的其他工具 | | |
| **方向** | PSTT->RF  RF->PSTT | | |
| **参数** | **名称** | **类型** | **参数说明** |
| source | 字符串 | 发生事件的模拟网元。  PSTT发送时，不携带该参数，source和office一致 |
| event | 字符串 | 事件内容的文本描述 |
| **编码举例** | PSTT->RF  {  "type": " Event",  "office": "enb1-mme1",  “event”: “omm command executed”,  }  RF->PSTT  {  "type": " Event",  "office": "enb2-mme2",  “source”: ”enb1-mme1”,  “event”: “omm command executed”,  } | | |
| **备注** | RF收到Event后，立即转发给所有其他的PSTT | | |

## Variable

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Variable | | |
| **用途1** | PSTT向RF上报变量 | | |
| **用途2** | RF将收到的变量广播给所有的其他工具 | | |
| **方向** | PSTT->RF  RF->PSTT | | |
| **参数** | **名称** | **类型** | **参数说明** |
| variable | 数组，复合型 | 变量描述 |
| variable.name | 字符串 | 变量名称 |
| Variable.type | 数值 | 变量类型 |
| variable.value | 字符串/数值 | 变量值 |
| **编码举例** | {  "type": " Variable",  "office": "enb1-mme1",  “variable”: {  [  {“name”: “var1”, “type”: <type\_code>, “value”: <value>}.  {“name”: “var2”, “type”: <type\_code>, “value”: <value>}  ]  }  } | | |
| **备注** | RF收到Variable后，立即转发给所有其他的PSTT  所有工具共享变量的名字空间 | | |



## Stop

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Stop | | |
| **用途** | RF要求PSTT停止测试流程 | | |
| **方向** | RF->PSTT | | |
| **参数** | **名称** | **类型** | **参数说明** |
| - | - | - |
| **编码举例** | {  "type": " Stop",  "office": "enb1-mme1",} | | |
| **备注** | 可以在任何时刻发送给其他测试工具。  如果脚本执行尚未开始工具就收到Stop，则上报不包含任何变量的Report，此时Report仅用于确认工具已经结束执行。 | | |



## Report

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Report | | |
| **用途** | PSTT向RF上报测试报告 | | |
| **方向** | PSTT->RF | | |
| **参数** | **名称** | **类型** | **参数说明** |
| report | 数组，复合类型 | 按照测试脚本要求采集的所有数据及其名称 |
| report[].name | 字符串 | 数据名称，与测试脚本中一致 |
| report[].value | 数值/字符串 | 数据的值 |
| **编码举例** | {  "type": " Report",  "office": "enb1-mme1",  “report”: [  {“name”: “var1”, “value”: <value1>},  {“name”: “var2”, “value”: <value2>}  ]  } | | |
| **备注** | 只能在发送了Started后发送，或者收到Stop后发送。 | | |

## Exception

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Report | | |
| **用途** | PSTT向RF上报异常 | | |
| **方向** | PSTT->RF | | |
| **参数** | **名称** | **类型** | **参数说明** |
| code | 数值 | 异常编号 |
| reason | 字符串 | 发生异常的原因 |
| **编码举例** | {  "type": " Exception",  "office": "enb1-mme1",  “code”: 1000  “reason”: “Script execution timeout.”  } | | |
| **备注** | 在各种异常场景下发送。 | | |

# PSTT测试脚本（Script）

## 脚本格式

测试脚本使用JSON格式编码的数据。

**{**

**“step”: <step type>,**

**<other step related parameters>**

**}**







## Send Message

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Send Message | | |
| **用途** | 执行到这一条时，PSTT向DUT发送一条消息 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| message | 复合型 | 消息 |
|  | message.layerN | 复合型 | 消息的一层 |
|  | message.layerN.protocol | 字符串 | 协议名称 |
|  | message.layerN.message | 字符串 | 消息名称 |
|  | message.layerN.parameters[] | 数组，复合型 | 消息的参数 |
|  | message.layerN.parameters[].name | 字符串 | 参数名称 |
|  | message.layerN.parameters[].value | 字符串 | 参数值 |
|  | message.layerN.para\_to\_save[] | 数组，复合型 | 需要保存的参数，结构与parameters一致 |
|  | message.layerN.para\_to\_retrieve[] | 数组，复合型 | 需要从变量的参数，结构与parameters一致 |
| **举例** | {  "step": "Send Message,  “message”: {  “layer1”: {  “protocol”: “SCCP”,  “message”: ConnectionRequest”,  “parameters”: [  {“name”: “para1”, “value”: “1”},  {“name”: “para2”, “value”: “2”},  ]  }  } | | |
| **备注** |  | | |

## Receive Message

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Receive Message | | |
| **用途** | 执行到这一条时，PSTT从DUT接收一条消息 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| message | 复合型 | 消息 |
|  | message.layerN | 复合型 | 消息的一层 |
|  | message.layerN.protocol | 字符串 | 协议名称 |
|  | message.layerN.message | 字符串 | 消息名称 |
|  | message.layerN.parameters[] | 数组，复合型 | 消息的参数 |
|  | message.layerN.parameters[].name | 字符串 | 参数名称 |
|  | message.layerN.parameters[].value | 字符串 | 参数值 |
|  | message.layerN.para\_to\_save[] | 数组，复合型 | 需要保存的参数，结构与parameters一致 |
|  | message.layerN.para\_to\_retrieve[] | 数组，复合型 | 需要从变量的参数，结构与parameters一致 |
| **举例** | {  "step": "Receive Message,  “message”: {  “layer1”: {  “protocol”: “SCCP”,  “message”: ConnectionRequest”,  “parameters”: [  {“name”: “para1”, “value”: “1”},  {“name”: “para2”, “value”: “2”},  ]  },  “index”: 1  } | | |
| **备注** |  | | |

## Send Event

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | SendEvent | | |
| **用途** | 执行到这一条时，PSTT向RF发送一个事件 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| event | 字符串 | 事件内容 |
| **举例** | {  "step": "Send Event",  “event”: “omm command success”  } | | |
| **备注** |  | | |

## Receive Event

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | SendEvent | | |
| **用途** | 执行到这一条时，PSTT等待接收RF转发的事件 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| source | 字符串 | 事件来源 |
| event | 字符串 | 事件内容 |
| timer | 数值 | 等待事件超时时间，0表示无限等待 |
| **举例** | {  "step": "Receive Event",  “source”: “enb1-mme1”,  “event”: “omm command success”,  “timer”: 10  } | | |
| **备注** |  | | |

## Wait

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Wait | | |
| **用途** | 执行到这一条时，PSTT等待指定的时间 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| timer | 数值 | 等待时间（s） |
| **举例** | {  "step": "Wait",  “timer”: 10  } | | |
| **备注** |  | | |

## Finish

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | Finish | | |
| **用途** | 执行到这一条时，PSTT结束脚本执行 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| **举例** | {  "step": "Finish",  } | | |
| **备注** |  | | |

## Retrieve Data

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | RetrieveData | | |
| **用途** | 执行这一条时，PSTT按照指定的操作方法，采集测试过程数据 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| operation | 字符串 | 采集数据的方法 |
| variable | 字符串 | 返回值的名称 |
| paras | 数组/复合型 | 操作需要的参数 |
| paras[].name | 字符串 | 参数名称，包含所有需要的参数 |
| paras[].value | 数值/字符串 | 参数值 |
| start\_message | 数组/复合型 | 获取数据的起始点，可选，不填写时从第一条消息开始 |
|  |  |  |
|  |  |  |
|  |  |  |
| start\_message[].name | 字符串 | 参数名称，包含message、index和其他参数 |
| start\_message [].value | 数值/字符串 | 参数值 |
|  |  |  |
| end\_message | 数组/复合型 | 获取数据的起始点，可选，不填写时从第一条消息开始 |
|  |  |  |
|  |  |  |
|  |  |  |
| start\_message[].name | 字符串 | 参数名称，包含message、index和其他参数 |
| start\_message [].value | 数值/字符串 | 参数值 |
|  |  |  |
| **举例** | {  “step”: “Retrieve Data”,  “variable”: “result”,  “operation”: “message count”,  “paras”: [{“name”: “message”, “value”: “AttachRsp”},  “].  “start\_message”: [{“name”: “message”, “value”: “AttachReq”},  {“name”, “index”, “value”: 3}],  “end\_message”: [{“name”: “message”, “value”: “AttachReq”},  {“name”, “index”, “value”: 5}]  } | | |
| **备注** |  | | |

## Set User Group

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | SetUserGroup | | |
| **用途** | 执行这一条时，PSTT选择指定的号码本 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| group | 字符串 | 号码本名称 |
| **举例** | {  “step”: “Set User Group”,  “group”: “subscriber group 1”  } | | |
| **备注** | RF保证这一条操作在脚本中所有业务消息前 | | |

## Set Strategy

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | SetStratergy | | |
| **用途** | 执行这一条时，PSTT设置指定的测试策略 | | |
| **参数** | **名称** | **类型** | **参数说明** |
| step | 字符串 | 操作类型 |
| config | 数组 | 策略 |
| config[].attribute | 字符串 | 策略名称 |
| config[].value | 字符串/数值 | 策略取值 |
| **举例** | {  “step”: “Set Strategy”,  “config”: [  {“attribute”: “subscriber mode”, “value”: “always the same subscriber”},  {“attribute”: “capture pcap file”, “value”: “only on failure”}  ]  } | | |
| **备注** | RF保证这一条操作在脚本中所有业务消息前 | | |

# Retrieve Data中各个操作的参数

## MessageCount

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **参数说明** |
| message | 字符串 | 消息名称 |
| 消息中的参数 | 字符串/数值 | 所有要符合条件的参数，可选。 |
|  |  |  |
|  |  |  |
| 返回值 | 数值 |  |

## MessageParameterCount

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **参数说明** |
| message | 字符串 | 消息名称 |
|  |  |  |
| 消息中的参数 | 字符串/数值 | 所有要符合条件的参数，可选。 |
|  |  |  |
|  |  |  |
| index | 数值 | 需要匹配第几个消息，不携带表示匹配所有。 |
| para\_to\_count | 字符串 | 参数名称 |
| 返回值 | 数值 |  |

## MessageParameterValue

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **参数说明** |
| message | 字符串 | 消息名称，全栈描述，由底向上，通过空格分隔 |
|  |  |  |
| 消息中的参数 | 字符串/数值 | 所有要符合条件的参数，可选。 |
|  |  |  |
|  |  |  |
| index | 数值 | 需要匹配第几个消息，不携带表示匹配所有。 |
| para\_to\_retrieve | 字符串 | 参数名称 |
| 返回值 | 数值/字符串 |  |

## MessageOrder

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **参数说明** |
| messages | 字符串 | 要查找的消息 |
| messages[].message | 字符串 | 消息名称，全栈描述，由底向上，通过空格分隔 |
|  |  |  |
| messages[].参数名称 | 复合型，数组 | 所有要符合条件的参数，可选。 |
|  |  |  |
|  |  |  |
| 返回值 | 数值 | 0-否，1-是 |

# 关键字参数与测试脚本参数的映射

## Send

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Send Message |
| message | message\_stack | message中有空格，视为stack，否则视为alias |
| message\_alias |
| paras='' | paras | 通过“,”分隔各个参数，每个参数书写为“name:value”的形式 |
| paras[].name |
| paras[].value |
| paras\_to\_retrieve | 通过“,”分隔各个参数，每个参数书写为“name:@var”的形式 |
| paras\_to\_retrieve [].name |
| paras\_to\_retrieve [].var |
| paras\_to\_save='' | paras\_to\_save | 通过“,”分隔各个参数，每个参数书写为“name:@var”的形式 |
| paras\_to\_save[].name |
| paras\_to\_save[].var |
| delay=‘0’ | delay |  |

## On Receive

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Receive Message |
| message | message\_stack | message中有空格，视为stack，否则视为alias |
| message\_alias |
| index=’0’ | index |  |
| paras='' | paras | 通过“,”分隔各个参数，每个参数书写为“name:value”的形式 |
| paras[].name |
| paras[].value |
| paras\_to\_save=‘’ | paras\_to\_save | 通过“,”分隔各个参数，每个参数书写为“name:@var”的形式 |
| paras\_to\_save[].name |
| paras\_to\_save[].var |

## Event

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Send Event/Receive Event |
| source | source | 仅用于Receive Event |
| event | event |  |
| timer | timer | 仅用于Receive Event |

## Wait

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Wait |
| timer | timer |  |

## Finish

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Finish |

## Retrieve Data

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Retrieve Data |
| result | variable |  |
| operation | operation |  |
| paras | paras | 通过“,”分隔各个参数，每个参数书写为“name:value”的形式 |
| start\_message='' | start\_message | 通过“,”分隔各个参数，每个参数书写为“name:value”的形式，其中，使用"name"表示消息名，使用"index"表示消息下标 |
| start\_message.alias |
| start\_message.stack |
| start\_message.paras |
| start\_message.paras[].name |
| start\_message.paras[].value |
| start\_message.index |
| end\_message='' | end\_message | 通过“,”分隔各个参数，每个参数书写为“name:value”的形式，其中，使用"name"表示消息名，使用"index"表示消息下标 |
| end\_message.alias |
| end\_message.stack |
| end\_message.paras |
| start\_message.paras[].name |
| start\_message.paras[].value |
| end\_message.index |

## Set User Group

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Set User Group |
| group | group |  |

## Set Strategy

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Script** | **说明** |
|  | step | Set Stragegy |
| config | config | 通过“,”分隔各个参数，每个参数书写为“attribute:value”的形式 |
| config[].attribute |
| config[].value |

# 测试结果异常

## 通信异常

通信异常由分发进程检测，当链路中断时，如果RF要求分发进程执行脚本，分发进程将通知RF，触发RF上报测试结果异常。

如果测试过程中，发现链路中断，分发进程通知RF，触发RF上报测试结果异常。

|  |  |
| --- | --- |
| **Exception Code** | **Exception Reason** |
| 1 | Link to test tools broken |

## 测试执行异常

工具在执行过程中检查下列异常，并通过Exception上报。

执行控制过程异常：

|  |  |
| --- | --- |
| **Exception Code** | **Exception Reason** |
| 1001 | Receive Start before script recieved |
| 1002 | Receive Event before script started |
| 1003 | Receive Varaible before script started |
| 1004 | Receive Stop before script started |
| 1005 | Script execution time out |
| 1006 | Wait event time out |
| 1007 | Set User Group after script has started |
| 1008 | Set Stratergy after script has started |
| 1009 | Using of undefined variable |
| 1010 | No action found for received message in script and knowledge base |

业务相关异常：

|  |  |
| --- | --- |
| **Exception Code** | **Exception Reason** |
| 2001 | Unknown message or parameter in Send |
| 2002 | Unknown message or parameter in Receive |
| 2003 | Received message cannot be decoded |
| 2004 | Unknown message or parameter in Retrieve Data |
| 2005 | Unknown operation in Retrieve Data |
| 2006 | Cannot find data to be retrieved |
| 2007 | Communication to DUT broken |

RF在执行过程中检测下列异常，并通过Exception上报。

|  |  |
| --- | --- |
| **Exception Code** | **Exception Reason** |
| 3001 | Receive unexpected message |
| 3002 | Link broken |
| 3003 | Time out |
|  |  |
|  |  |
|  |  |
|  |  |

# 状态机

## RF



## PSTT

