# 环境准备

## 提取BPS TCL库

参阅文档“Extract BPS Libs.docx”提取本地匹配版本的BPS TCL库

## 准备本地自动化环境

将步骤1.1中提取的BPS库文件Copy到ixia-bps-api\Bps，并确保调用脚本之前使用了命令：lappend auto\_path ./ ixia-bps-api 将IxiaBpsTester加进TCL环境。

注释：IxiaBps可以从Github上拿到：https://github.com/jundong/ixia-bps-api

# 构建对象

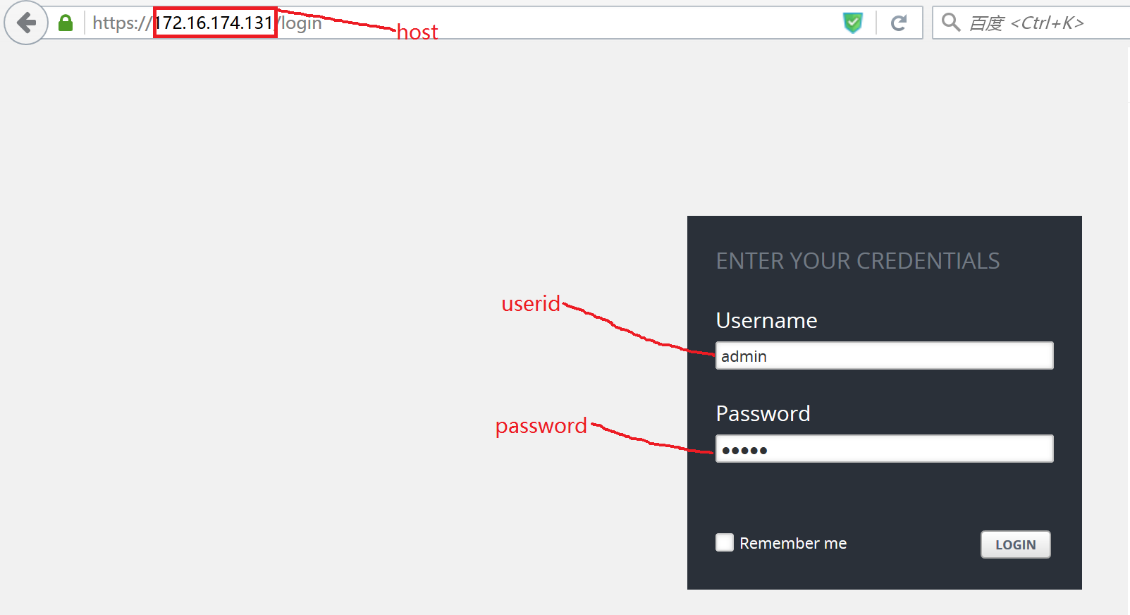
## 构建Tester对象语法

|  |
| --- |
|  |

**Tester @tester host userid password args**

### 参数

|  |
| --- |
|  |



* args：详细内容请参阅手册1145页

-name: Allows you to name or rename an item

-onclose: Determines what a script does once it finishes

Running

-shortcuts: Set the value to ‘true’ to enable shortcut

commands for test components. The default value is “true”

### 示例

|  |
| --- |
|  |

假设你的BPS库文件保存在”c:/ ixia-bps-api”目录下:

**lappend auto\_path “c:/ ixia-bps-api”**

**package require IxiaBps**

**namespace import IXIA::\***

**Tester @tester 172.16.174.131 admin admin –shortcuts false**

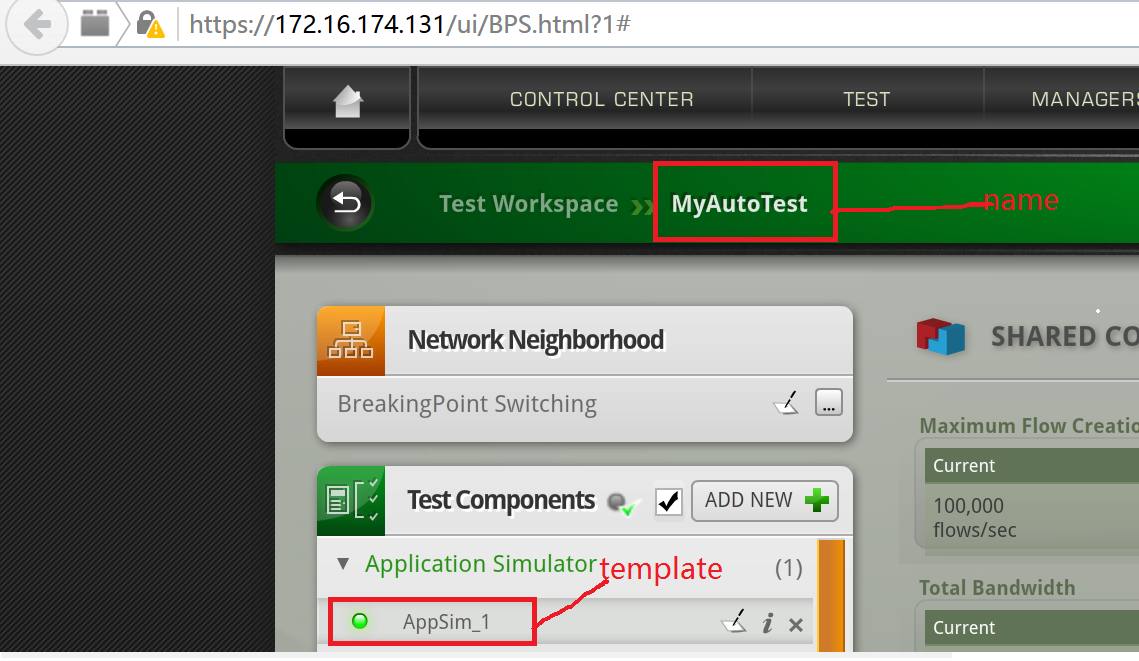
## 构建Test: createTest

|  |
| --- |
|  |

**@tester createTest name args**

### 参数

|  |
| --- |
|  |



* name：Test 名字
* args：参数如下：

-template: 指定创建Test时的component，系统预定义可选参数请参阅手册1249-1252页，如果使用已有测试或自定义Test为模板创建新Test，直接指定已有Test名字即可

### 示例

|  |
| --- |
|  |

**@tester createTest $testName –template “AppSim”**

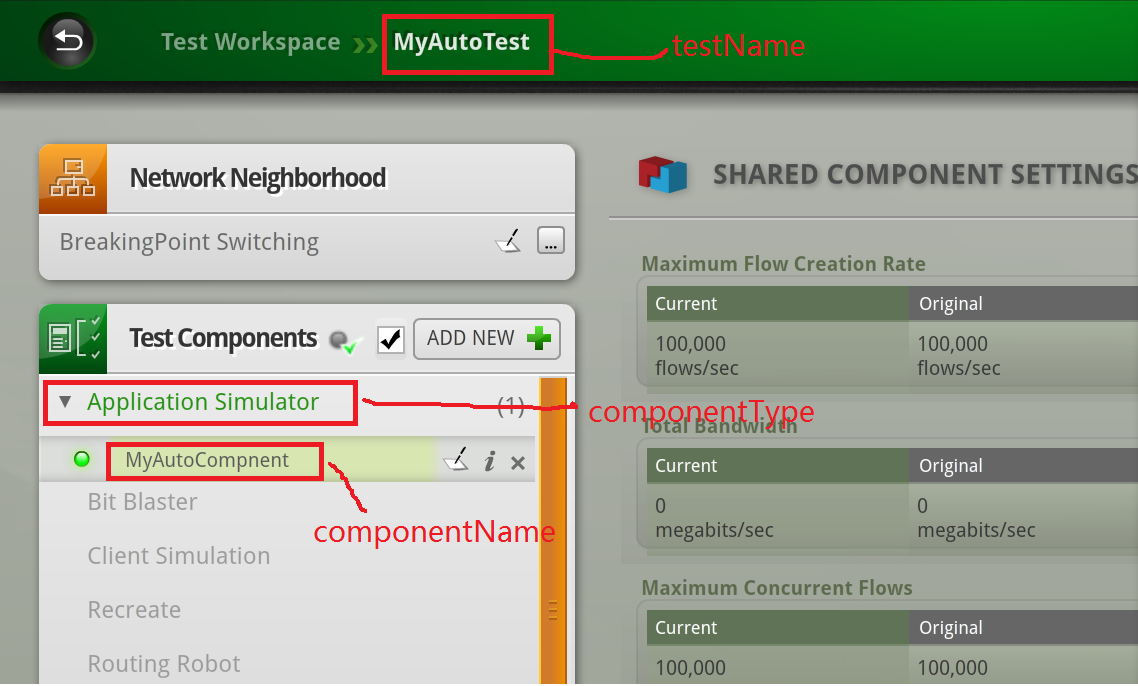
## 构建Component: createComponent

|  |
| --- |
|  |

**@tester createComponent testName componentName componentType args**

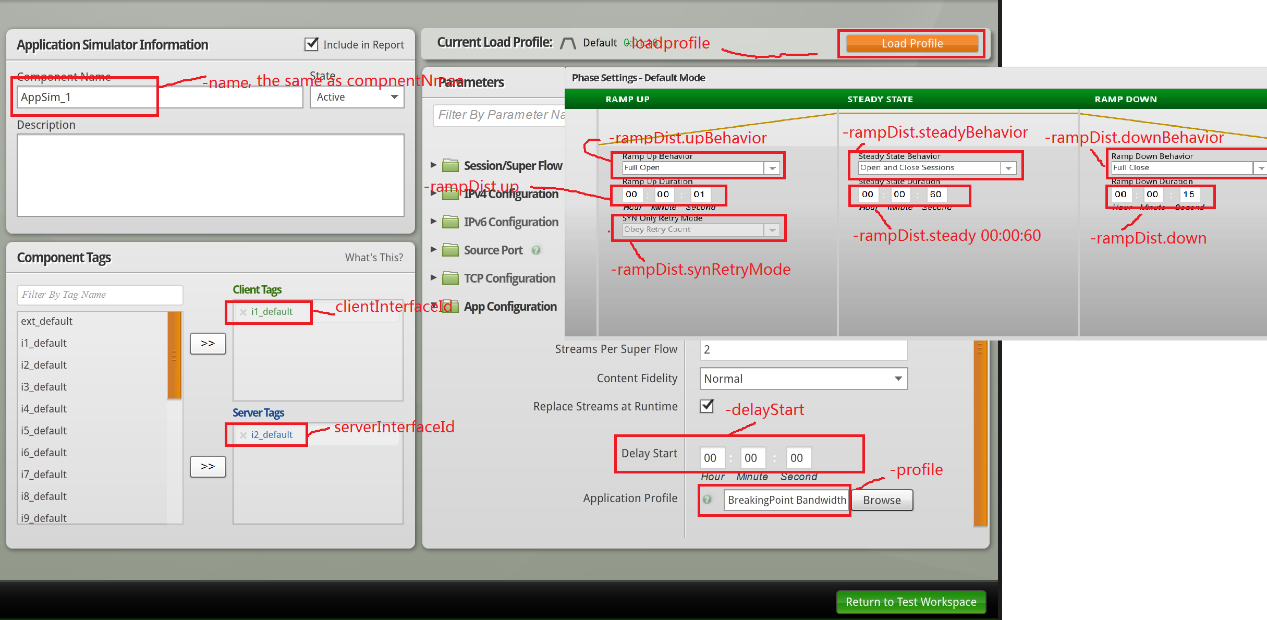
### 参数

|  |
| --- |
|  |



* testName：Test 名字
* componentName：Component 名字
* componentType：Component 类型，例如：appsim…
* args：有二个必选参数：clientInterfaceId 和serverInterfaceId，其它可选参数可以通过Component对象的configure方法列举出来

常用args与界面的对应关系：



{-app.fidelity {} {}} {-app.replace\_streams {} {}} {-app.streamsPerSuperflow 2 2} {-client\_tags i1\_default i1\_default} {-delayStart 00:00:00 00:00:00} {-description {A reasonable starting-point for a custom test。}} {-ip.tos 0 0} {-ip.ttl 32 32} {-ip6.flowlabel 0 0} {-ip6.hop\_limit 64 64} {-ip6.traffic\_class 0 0} {-loadprofile None None} {-name MyAutoComponent MyAutoComponent} {-profile {BreakingPoint Bandwidth} {BreakingPoint Bandwidth}} {-rampDist.down 00:00:15 00:00:15} {-rampDist.downBehavior full full} {-rampDist.steady 00:00:60 00:00:60} {-rampDist.steadyBehavior cycle cycle} {-rampDist.synRetryMode obey\_retry obey\_retry} {-rampDist.up 00:00:01 00:00:01} {-rampDist.upBehavior full full} {-rampUpProfile.increment {} {}} {-rampUpProfile.interval {} {}} {-rampUpProfile.max {} {}} {-rampUpProfile.min {} {}} {-rampUpProfile.type calculated calculated} {-rateDist.max {} {}} {-rateDist.min 10000 10000} {-rateDist.scope aggregate aggregate} {-rateDist.type constant constant} {-rateDist.unit mbps mbps} {-rateDist.unlimited false false} {-server\_tags i2\_default i2\_default} {-sessions.allocationOverride {} {}} {-sessions.closeFast true true} {-sessions.emphasis {} {}} {-sessions.engine {} {}} {-sessions.max 100000 100000} {-sessions.maxActive 0 0} {-sessions.maxPerSecond 100000 100000} {-sessions.openFast false false} {-sessions.statDetail {} {}} {-sessions.target 1 1} {-sessions.targetMatches {} {}} {-sessions.targetPerSecond 1 1} {-srcPortDist.max 65535 65535} {-srcPortDist.min 1024 1024} {-srcPortDist.type random random} {-tcp.add\_timestamps true true} {-tcp.aging\_time 0 0} {-tcp.aging\_time\_data\_type {} {}} {-tcp.delay\_acks true true} {-tcp.ecn {} {}} {-tcp.handshake\_data false false} {-tcp.initial\_congestion\_window {} {}} {-tcp.initial\_receive\_window 5792 5792} {-tcp.mss 1460 1460} {-tcp.raw\_flags {} {}} {-tcp.reset\_at\_end false false} {-tcp.retries 3 3} {-tcp.retry\_quantum\_ms 500 500} {-tcp.shutdown\_data {} {}} {-tcp.tcp\_connect\_delay\_ms {} {}} {-tcp.tcp\_keepalive\_timer {} {}} {-tcp.tcp\_window\_scale {} {}}

### 示例

|  |
| --- |
|  |

**@tester createComponent $testNewName $componentName "appsim" 1 2 -rampDist.down 00:00:15**

**set component [ @tester getComponent $componentName ]**

**#下面列举createComponent时的可选参数**

**$ component configure**

{-app.fidelity {} {}} {-app.replace\_streams {} {}} {-app.streamsPerSuperflow 2 2} {-client\_tags i1\_default i1\_default} {-delayStart 00:00:00 00:00:00} {-description {A reasonable starting-point for a custom test.}} {-ip.tos 0 0} {-ip.ttl 32 32} {-ip6.flowlabel 0 0} {-ip6.hop\_limit 64 64} {-ip6.traffic\_class 0 0} {-loadprofile None None} {-name MyAutoComponent MyAutoComponent} {-profile {BreakingPoint Bandwidth} {BreakingPoint Bandwidth}} {-rampDist.down 00:00:15 00:00:15} {-rampDist.downBehavior full full} {-rampDist.steady 00:00:60 00:00:60} {-rampDist.steadyBehavior cycle cycle} {-rampDist.synRetryMode obey\_retry obey\_retry} {-rampDist.up 00:00:01 00:00:01} {-rampDist.upBehavior full full} {-rampUpProfile.increment {} {}} {-rampUpProfile.interval {} {}} {-rampUpProfile.max {} {}} {-rampUpProfile.min {} {}} {-rampUpProfile.type calculated calculated} {-rateDist.max {} {}} {-rateDist.min 10000 10000} {-rateDist.scope aggregate aggregate} {-rateDist.type constant constant} {-rateDist.unit mbps mbps} {-rateDist.unlimited false false} {-server\_tags i2\_default i2\_default} {-sessions.allocationOverride {} {}} {-sessions.closeFast true true} {-sessions.emphasis {} {}} {-sessions.engine {} {}} {-sessions.max 100000 100000} {-sessions.maxActive 0 0} {-sessions.maxPerSecond 100000 100000} {-sessions.openFast false false} {-sessions.statDetail {} {}} {-sessions.target 1 1} {-sessions.targetMatches {} {}} {-sessions.targetPerSecond 1 1} {-srcPortDist.max 65535 65535} {-srcPortDist.min 1024 1024} {-srcPortDist.type random random} {-tcp.add\_timestamps true true} {-tcp.aging\_time 0 0} {-tcp.aging\_time\_data\_type {} {}} {-tcp.delay\_acks true true} {-tcp.ecn {} {}} {-tcp.handshake\_data false false} {-tcp.initial\_congestion\_window {} {}} {-tcp.initial\_receive\_window 5792 5792} {-tcp.mss 1460 1460} {-tcp.raw\_flags {} {}} {-tcp.reset\_at\_end false false} {-tcp.retries 3 3} {-tcp.retry\_quantum\_ms 500 500} {-tcp.shutdown\_data {} {}} {-tcp.tcp\_connect\_delay\_ms {} {}} {-tcp.tcp\_keepalive\_timer {} {}} {-tcp.tcp\_window\_scale {} {}}

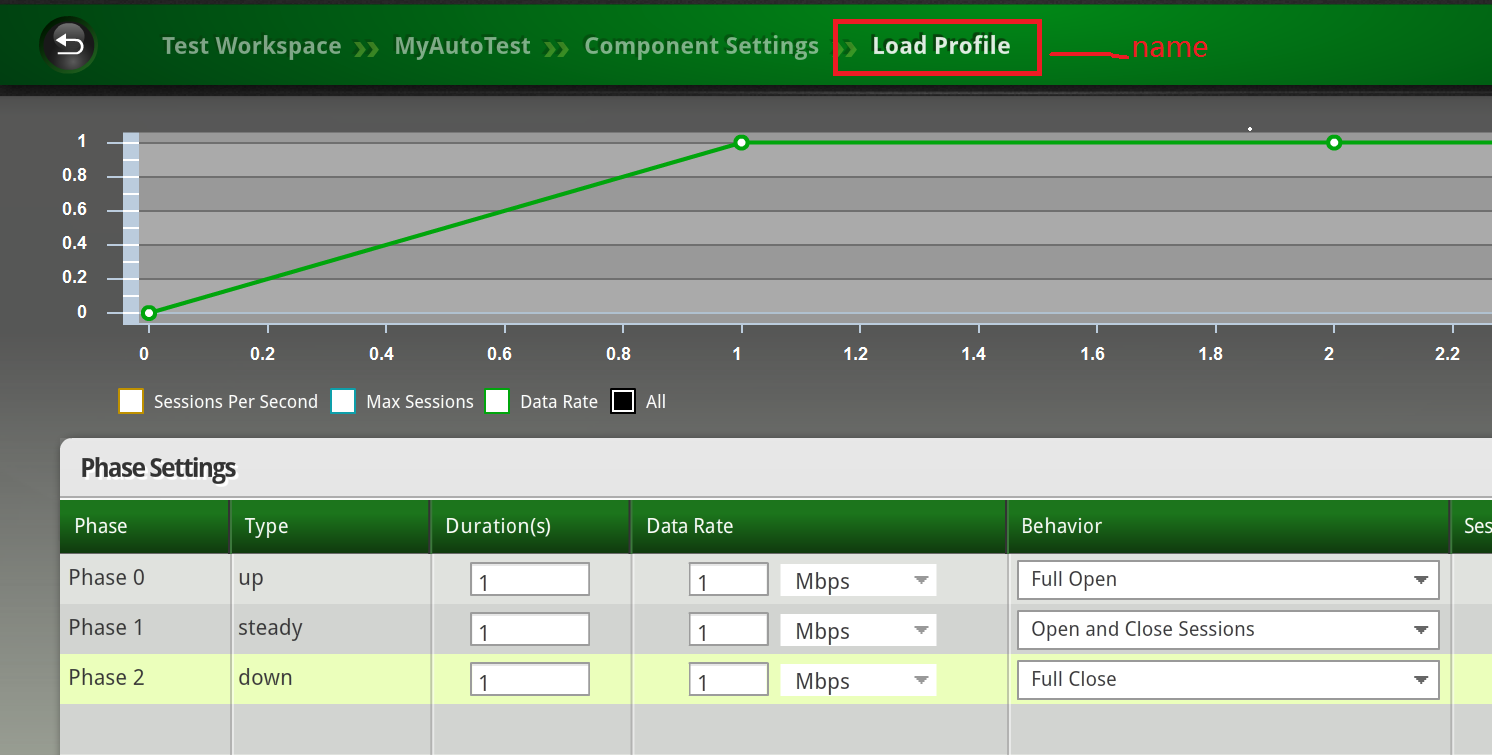
## 构建用户自定义LoadProfile: createLoadProfile

|  |
| --- |
|  |

**@tester createLoadProfile name args**

### 参数

|  |
| --- |
|  |



* name：自定义LoadProfile名字
* args：

–name：第一个参数name相同，会被第一个参数name覆盖

–template：不指定即创建系统默认指定的LoadProfile，指定以存在的LoadProfile为模板创建LoadProfile

### 示例

|  |
| --- |
|  |

**@tester createLoadProfile $newLoadProfileName -template $loadProfileName**

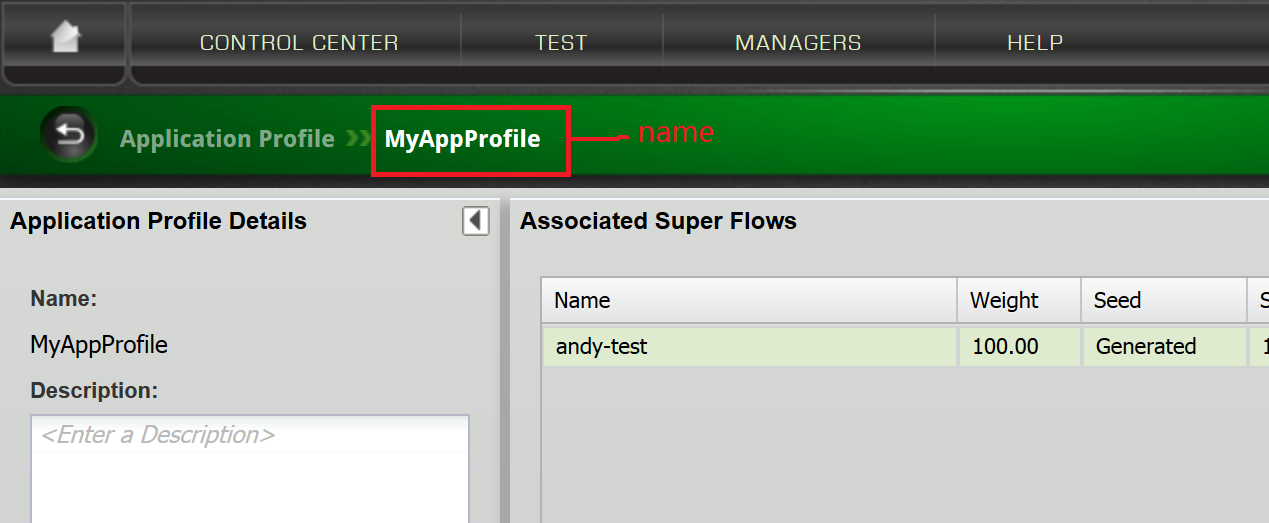
## 构建AppProfile: createAppProfile

|  |
| --- |
|  |

**@tester createAppProfile name args**

### 参数

|  |
| --- |
|  |



* name：AppProfile名字
* args：

–name：第一个参数name相同，会被第一个参数name覆盖

–template：不指定即创建系统默认的AppProfile，指定以存在的AppProfile为模板创建AppProfile

### 示例

|  |
| --- |
|  |

**@tester createAppProfile $newAppProfile -template $appProfile**

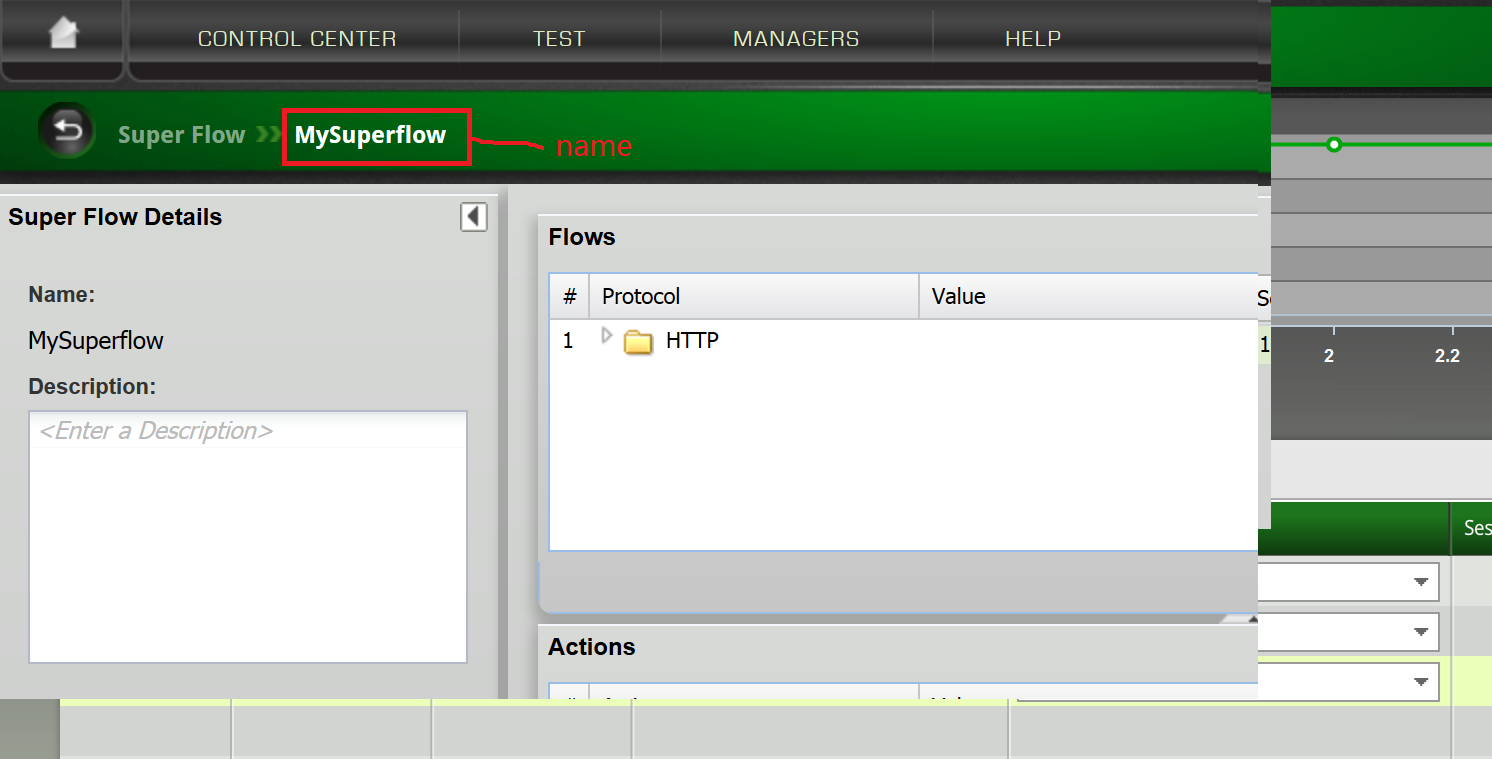
## 构建Superflow: createSuperflow

|  |
| --- |
|  |

**@tester createSuperflow name args**

### 参数

|  |
| --- |
|  |



* name：Superflow名字
* args：

–name：第一个参数name相同，会被第一个参数name覆盖

–template：不指定即创建系统默认的Superflow，指定以存在的Superflow为模板创建Superflow

### 示例

|  |
| --- |
|  |

**@tester createSuperflow $newSuperflow -template $superflow**

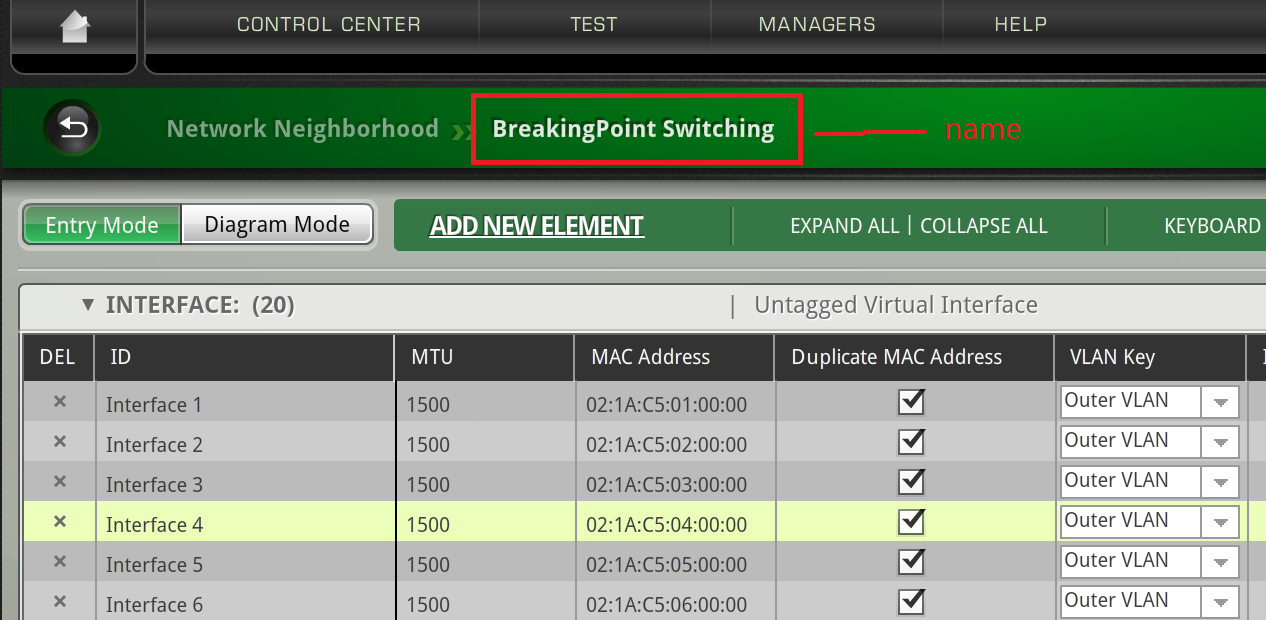
## 构建Network: createNetwork

|  |
| --- |
|  |

**@tester createNetwork name args**

### 参数

|  |
| --- |
|  |



* name：Network名字
* args：

–name：第一个参数name相同，会被第一个参数name覆盖

–template：不指定即创建系统默认的Network，指定以存在的Network为模板创建Network

### 示例

|  |
| --- |
|  |

**@tester createNetwork $newNetwork -template $network**

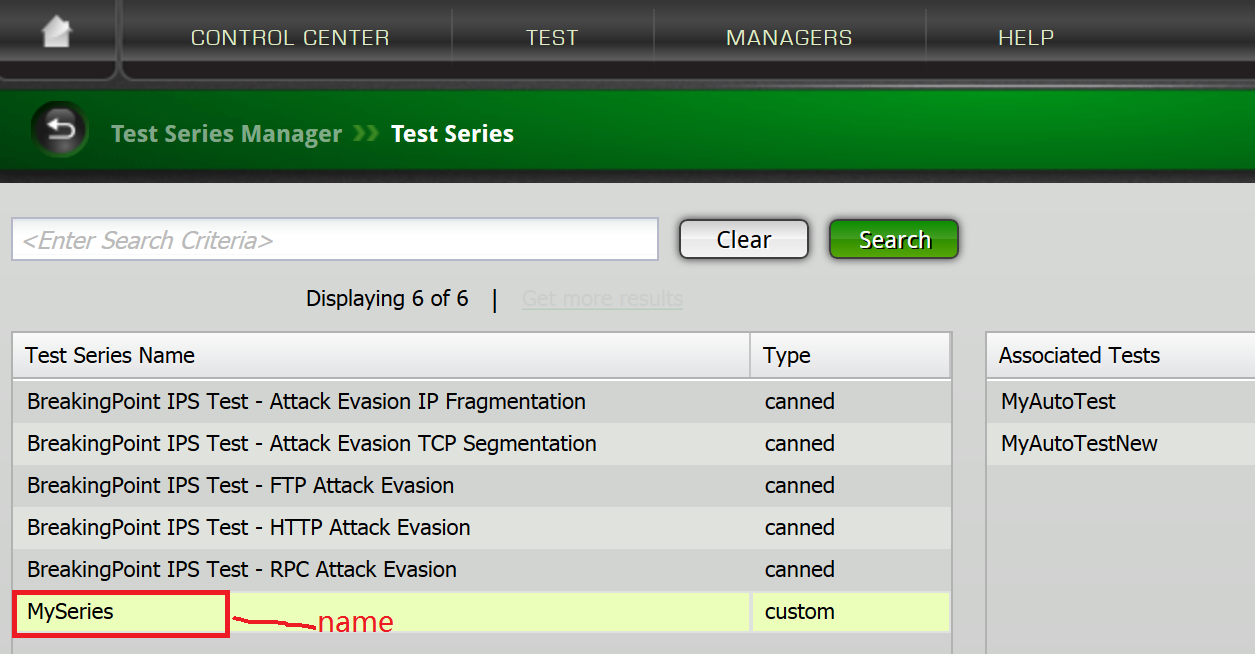
## 构建TestSeries: createTestSeries

|  |
| --- |
|  |

**@tester createTestSeries name args**

### 参数

|  |
| --- |
|  |



* name：TestSeries名字
* args：

–name：第一个参数name相同，会被第一个参数name覆盖

–template：不指定即创建系统默认的TestSeries，指定以存在的TestSeries为模板创建TestSeries

### 示例

|  |
| --- |
|  |

**@tester createTestSeries $newNetwork -template $network**

# 配置对象

## 全局配置方法：configure

|  |
| --- |
|  |

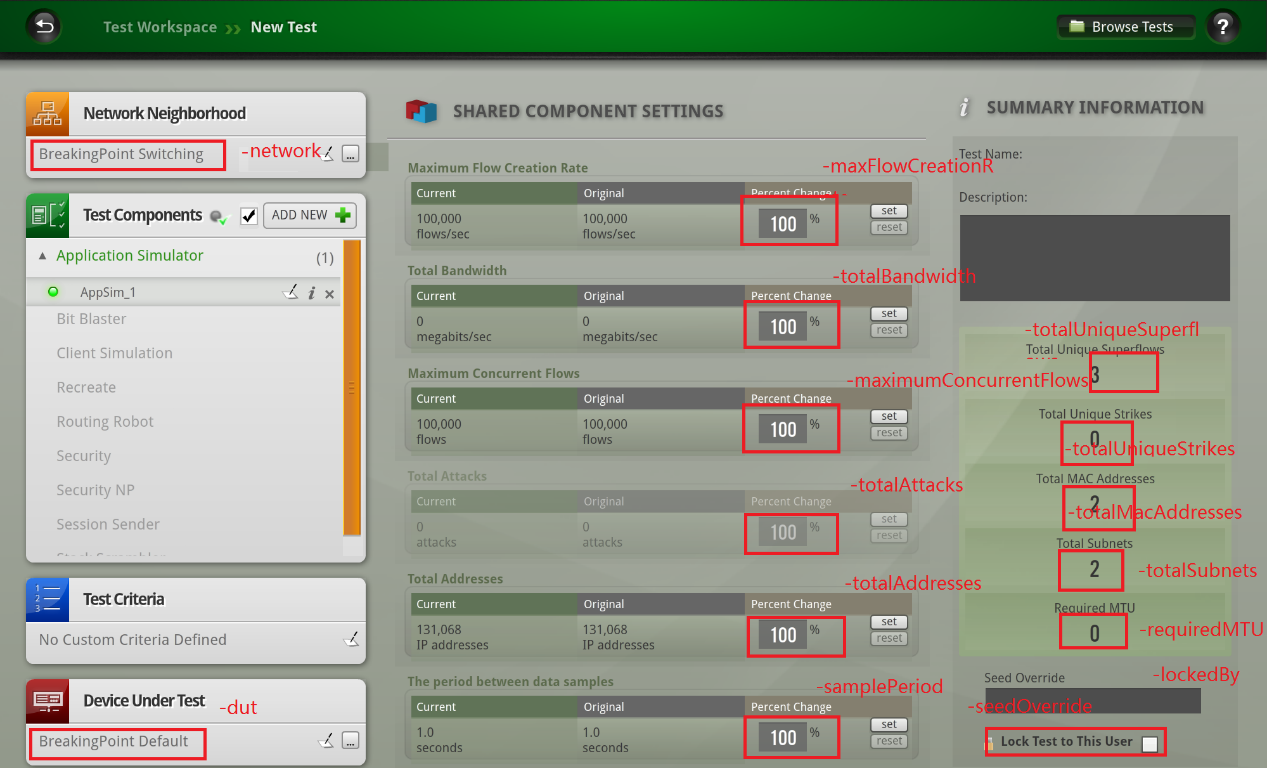
**@tester configure name args**

### 参数

|  |
| --- |
|  |

* name：构建的对象的名字，可以是Test， TestSeries，LoadProfile，Network，AppProfile， Superflow和Component名字
* args：对应对象的参数列表，具体描述如下：
  + Test 可配置参数如下:

{-description {} {}} {-dut {BreakingPoint Default} {BreakingPoint Default}} {-lockedBy {} {}} {-maxFlowCreationRate 100 100} {-maximumConcurrentFlows 100 100} {-name MyAutoTest MyAutoTest} {-neighborhood {BreakingPoint Switching} {BreakingPoint Switching}} {-network {BreakingPoint Switching} {BreakingPoint Switching}} {-requiredMTU 0 0} {-samplePeriod 100 100} {-seedOverride {} {}} {-totalAddresses 100 100} {-totalAttacks 100 100} {-totalBandwidth 100 100} {-totalMacAddresses 0 2} {-totalSubnets 0 2} {-totalUniqueStrikes 0 0} {-totalUniqueSuperflows 0 3}

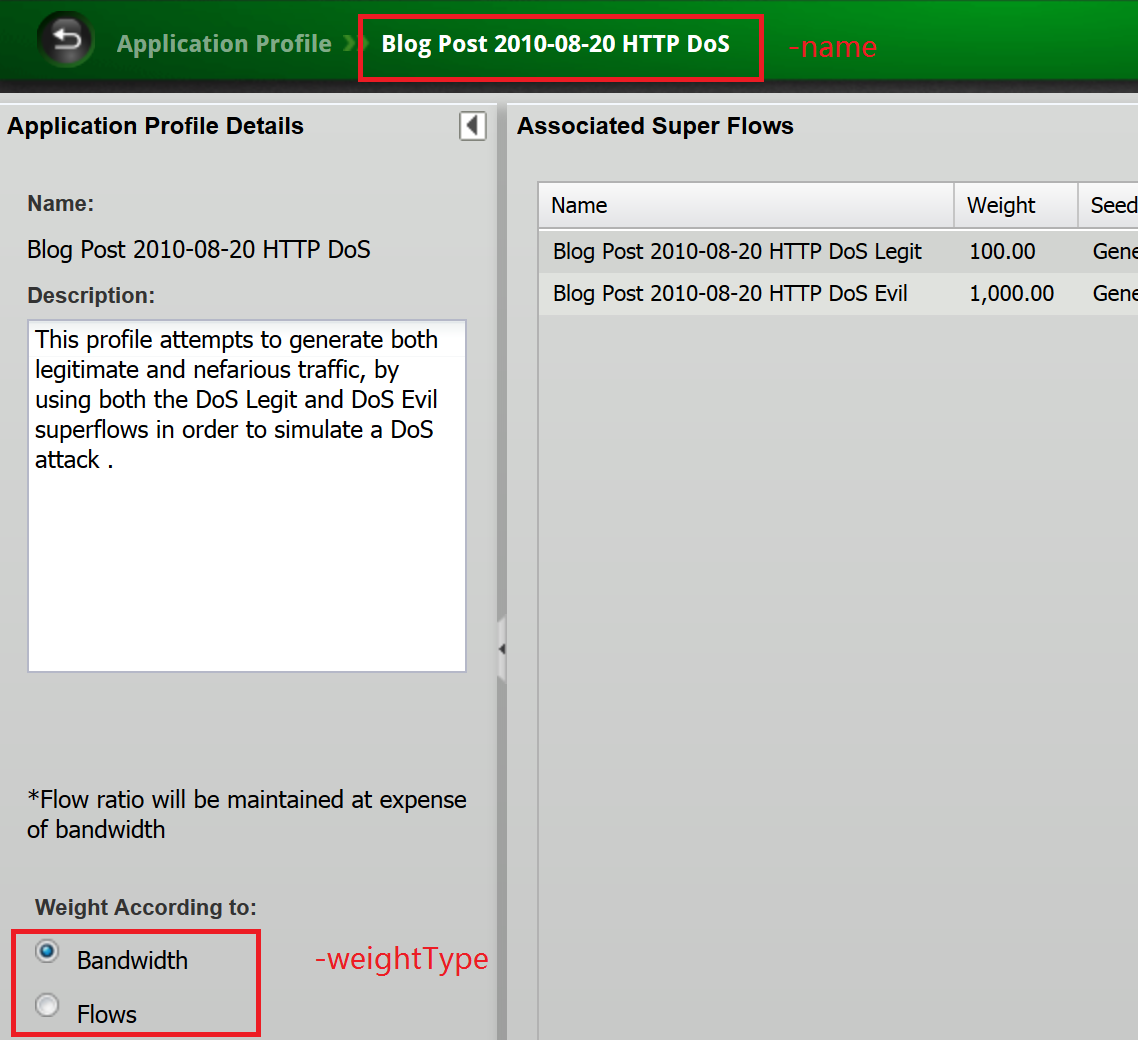


* + TestSeries 可配置参数如下:

{-name ts1 ts1}

* + AppProfile 可配置参数如下:

{-name MyAppProfile MyAppProfile} {-weightType bandwidth bandwidth}



* + Network 可配置参数如下:

{-name {BreakingPoint Switching} {BreakingPoint Switching}}

* + LoadProfile 可配置参数如下:

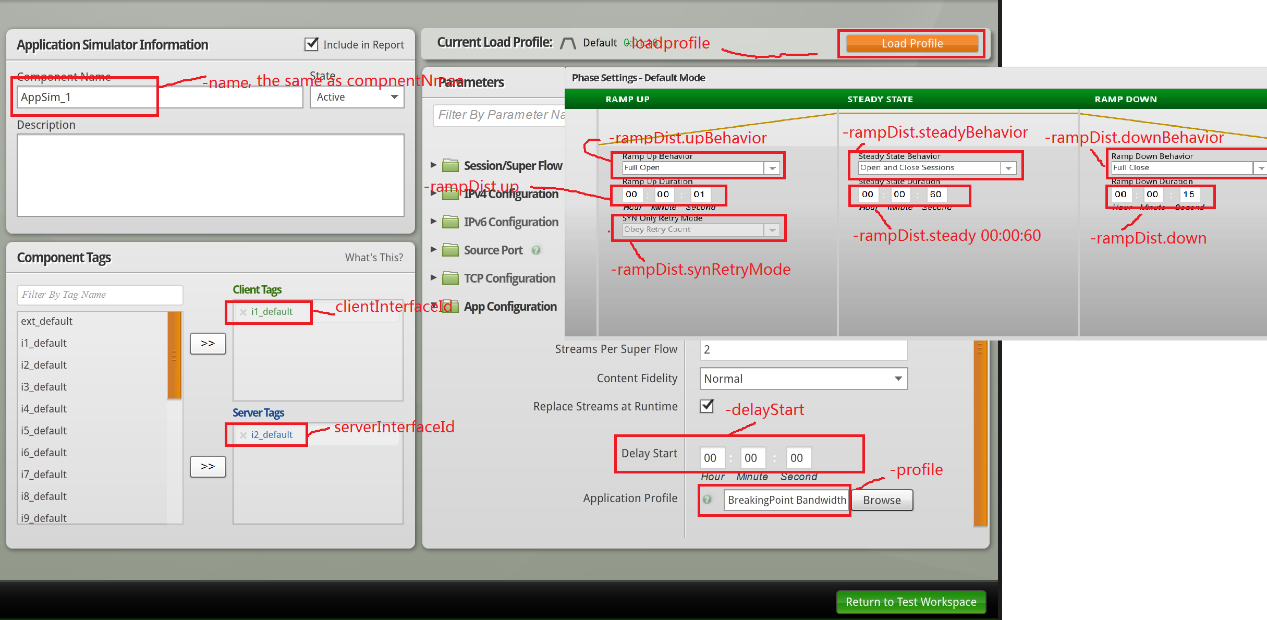
{-name {BreakingPoint Default} MyAutoLoadProfile}

* + Superflow可配置参数如下:

{-name MySuperflow MySuperflow}

* + Component 可配置参数如下:

{-app.fidelity {} {}} {-app.replace\_streams {} {}} {-app.streamsPerSuperflow 2 2} {-client\_tags i1\_default i1\_default} {-delayStart 00:00:00 00:00:00} {-description {A reasonable starting-point for a custom test. }} {-ip.tos 0 0} {-ip.ttl 32 32} {-ip6.flowlabel 0 0} {-ip6.hop\_limit 64 64} {-ip6.traffic\_class 0 0} {-loadprofile None None} {-name MyAutoComponent MyAutoComponent} {-profile {BreakingPoint Bandwidth} {BreakingPoint Bandwidth}} {-rampDist.down 00:00:15 00:00:15} {-rampDist.downBehavior full full} {-rampDist.steady 00:00:60 00:00:60} {-rampDist.steadyBehavior cycle cycle} {-rampDist.synRetryMode obey\_retry obey\_retry} {-rampDist.up 00:00:01 00:00:01} {-rampDist.upBehavior full full} {-rampUpProfile.increment {} {}} {-rampUpProfile.interval {} {}} {-rampUpProfile.max {} {}} {-rampUpProfile.min {} {}} {-rampUpProfile.type calculated calculated} {-rateDist.max {} {}} {-rateDist.min 10000 10000} {-rateDist.scope aggregate aggregate} {-rateDist.type constant constant} {-rateDist.unit mbps mbps} {-rateDist.unlimited false false} {-server\_tags i2\_default i2\_default} {-sessions.allocationOverride {} {}} {-sessions.closeFast true true} {-sessions.emphasis {} {}} {-sessions.engine {} {}} {-sessions.max 100000 100000} {-sessions.maxActive 0 0} {-sessions.maxPerSecond 100000 100000} {-sessions.openFast false false} {-sessions.statDetail {} {}} {-sessions.target 1 1} {-sessions.targetMatches {} {}} {-sessions.targetPerSecond 1 1} {-srcPortDist.max 65535 65535} {-srcPortDist.min 1024 1024} {-srcPortDist.type random random} {-tcp.add\_timestamps true true} {-tcp.aging\_time 0 0} {-tcp.aging\_time\_data\_type {} {}} {-tcp.delay\_acks true true} {-tcp.ecn {} {}} {-tcp.handshake\_data false false} {-tcp.initial\_congestion\_window {} {}} {-tcp.initial\_receive\_window 5792 5792} {-tcp.mss 1460 1460} {-tcp.raw\_flags {} {}} {-tcp.reset\_at\_end false false} {-tcp.retries 3 3} {-tcp.retry\_quantum\_ms 500 500} {-tcp.shutdown\_data {} {}} {-tcp.tcp\_connect\_delay\_ms {} {}} {-tcp.tcp\_keepalive\_timer {} {}} {-tcp.tcp\_window\_scale {} {}}



### 示例

|  |
| --- |
|  |

**@tester configure $newNetwork –name “changedNetworkName”**

## 配置LoadPhase方法：configurePhase

|  |
| --- |
|  |

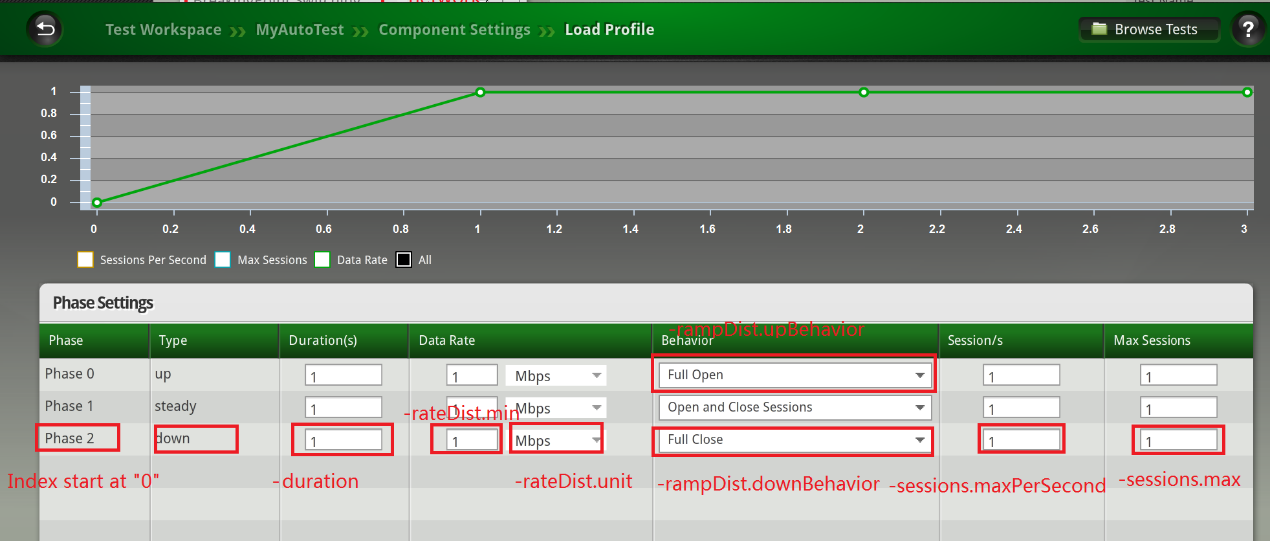
**@tester configurePhase name action index args**

### 参数

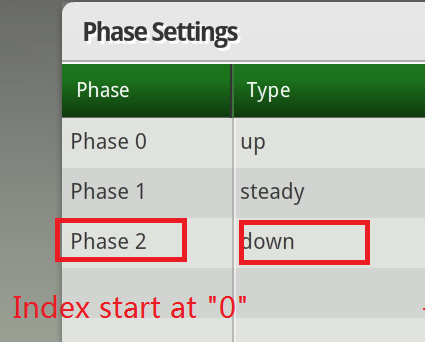
|  |
| --- |
|  |

* name：LoadProfile名字
* action：可以是add：添加一个Phase，modify：修改一个Phase和remove：删除一个Phase
* index：指定要修改的Phase
* args：
  + action取add时的可选参数：

0 {-duration 1 -rampDist.upBehavior full -sessions.max 50000 -sessions.maxPerSecond 50000 -rateDist.min 900 -rateDist.unit mbps -rateDist.scope per\_if} 1 {-duration 28 -rampDist.steadyBehavior cycle -sessions.max 50000 -sessions.maxPerSecond 50000 -rateDist.min 900 -rateDist.unit mbps -rateDist.scope per\_if} 2 {-duration 1 -rampDist.downBehavior full -sessions.max 1 -sessions.maxPerSecond 50000 -rateDist.min 900 -rateDist.unit mbps -rateDist.scope per\_if}



* + action取modify时的可选参数和add相同
  + action取remove时的唯一参数是index，即Phase从“0”开始的index：



### 示例

|  |
| --- |
|  |

**set action "add"**

**set index 2**

**@tester configurePhase $newLoadProfile $action $index -duration 11 \**

**-rampDist.steadyBehavior cycle \**

**-sessions.max 11111 \**

**-sessions.maxPerSecond 11111 \**

**-rateDist.min 111 \**

**-rateDist.unit mbps \**

**-rateDist.scope per\_if**

**set action "modify"**

**set index 2**

**@tester configurePhase $newLoadProfile $action $index -duration 222 \**

**-rampDist.steadyBehavior cycle \**

**-sessions.max 22222 \**

**-sessions.maxPerSecond 22222 \**

**-rateDist.min 2222 \**

**-rateDist.unit mbps \**

**-rateDist.scope per\_if**

**set action "remove"**

**set index 2**

**@tester configurePhase $newLoadProfile $action $index**

## 配置Network方法：configureNetwork

|  |
| --- |
|  |

**@tester configureNetwork name action type parameters args**

### 参数

|  |
| --- |
|  |

* name：Network名字
* action：可以是add：添加一个Element，config：配置一个Element和remove：删除一个Element
* type：查看可配置的Element，可以通过如下方法：

@tester createNetwork $newNetwork -template $network

set network [ @tester getNetwork $newNetwork ]

$network elementTypes

interface {label Interface category {IP Infrastructure} description {Untagged Virtual Interface}} vlan {label VLAN category {IP Infrastructure} description {Virtual Interface}} ip\_dhcp\_server {label {IPv4 DHCP Server} category {IP Infrastructure} description {Simulated DHCP server}} ip6\_dhcp\_server {label {IPv6 DHCP Server} category {IP Infrastructure} description {Simulated DHCPv6 server}} ip\_router {label {IPv4 Router} category {IP Infrastructure} description {Simulated IPv4 router}} ipsec\_router {label {IPsec IPv4 Router} category {IP Infrastructure} description {Simulated IPsec IPv4 router}} ip6\_router {label {IPv6 Router} category {IP Infrastructure} description {Simulated IPv6 router}} ip\_dns\_config {label {IPv4 DNS Configuration} category {IP Configuration} description {Shared DNS configuration for IPv4 endpoints}} ip6\_dns\_config {label {IPv6 DNS Configuration} category {IP Configuration} description {Shared DNS configuration for IPv6 endpoints}} ipsec\_config {label {IPsec Configuration} category {IP Configuration} description {IPsec Configuration}} ip\_external\_hosts {label {IPv4 External Hosts} category Endpoint description {External hosts used as a test target}} ip6\_external\_hosts {label {IPv6 External Hosts} category Endpoint description {External hosts used as a test target}} ip\_static\_hosts {label {IPv4 Static Hosts} category Endpoint description {Simulated IPv4 endpoints}} ip6\_static\_hosts {label {IPv6 Hosts} category Endpoint description {Simulated IPv6 endpoints}} ip\_dhcp\_hosts {label {IPv4 DHCP Hosts} category Endpoint description {Simulated DHCP endpoints}} sixrd\_ce {label {6RD Customer Edge Routers} category Endpoint description {Simulated 6RD Customer Edge Routers}} ue {label {User Equipment} category Endpoint description {Devices that transmit data over a 3G or LTE mobile network}} enodeb\_mme {label {eNodeB/MME (GTPv2)} category LTE description {Simulates eNodeB/MME S1-U and S11 interfaces}} enodeb\_mme6 {label {eNodeB/MME IPv6 (GTPv2)} category LTE description {Simulates eNodeB/MME S1-U and S11 interfaces IPv6}} enodeb\_mme\_sgw {label {eNodeB/MME/SGW (GTPv2)} category LTE description {Simulates SGW S5/S8 interface w/ eNodeB and MME}} enodeb\_mme\_sgw6 {label {eNodeB/MME/SGW IPv6(GTPv2)} category LTE description {Simulates SGW S5/S8 interface w/ eNodeB and MME IPv6}} enodeb {label {eNodeB (S1AP / GTPv1)} category LTE description {Simulates a set of eNodeBs emulating S11 and S1-U interfaces}} enodeb6 {label {eNodeB IPv6(S1AP / GTPv1)} category LTE description {Simulates a set of eNodeBs emulating S11 and S1-U interfaces using IPv6}} sgw\_pgw {label SGW/PGW category LTE description {Serving Gateway/PDN Gateway}} sgw\_pgw6 {label {SGW/PGW IPv6} category LTE description {Serving Gateway/PDN Gateway IPv6}} mme\_sgw\_pgw {label MME/SGW/PGW category LTE description {Mobile Management Entity/Serving Gateway/PDN Gateway}} mme\_sgw\_pgw6 {label {MME/SGW/PGW IPv6} category LTE description {Mobile Management Entity/Serving Gateway/PDN Gateway IPv6}} pgw {label PGW category LTE description {Simulates the PDN Gateway S5/S8 Interface}} pgw6 {label {PGW IPv6} category LTE description {Simulates the PDN Gateway S5/S8 Interface IPv6}} ue\_info {label {HSS/UE database} category {Mobile Configuration} description {Information required to properly represent User Equipment on an LTE mobile network}} plmn {label {Public Land Mobile Network} category {Mobile Configuration} description {A regulatory domain for a mobile network}} mobility\_session\_info {label {Mobility Session Information} category {Mobile Configuration} description {Information required for a device to connect to a mobile network}} ggsn {label GGSN category 3G description {Gateway GPRS Support Node}} ggsn6 {label {GGSN IPV6} category 3G description {Gateway GPRS Support Node IPV6}} sgsn {label SGSN category 3G description {Service GPRS Support Node}} sgsn6 {label {SGSN IPv6} category 3G description {Service GPRS Support Node IPv6}} ds\_lite\_b4 {label {DS-Lite B4} category {IP Infrastructure} description {IPv6-aware CPE with a B4 interface}} ds\_lite\_aftr {label {DS-Lite AFTR} category {IP Infrastructure} description {AFTR Router}} slaac\_cfg {label {IPv6 SLAAC Client Configuration} category {IP Configuration} description {Shared IPv6 SLAAC Client Configuration}} dhcpv6c\_cfg {label {DHCPv6 Client Configuration} category {IP Configuration} description {Shared DHCPv6 Client Configuration}} dhcpv6c\_req\_opts\_cfg {label {DHCPv6 Request Options Configuration} category {IP Configuration} description {DHCPv6 Request Options}} dhcpv6c\_tout\_and\_retr\_cfg {label {DHCPv6 Timeout and Retransmission} category {IP Configuration} description {DHCPv6 Timing Options}}

* parameters：根据不同Element类型，给予不同的参数列表：
  + action为add，并且type为interface时，parameters为空参数列表
  + action为add，并且type为path时，parameters为[list endpointid1 endpointid2]参数列表
  + action为config，并且type为interface时，parameters为[ list NewInterfaceId]参数列表
  + action为remove，并且type为interface时，parameters为[ list NewInterfaceId]参数列表

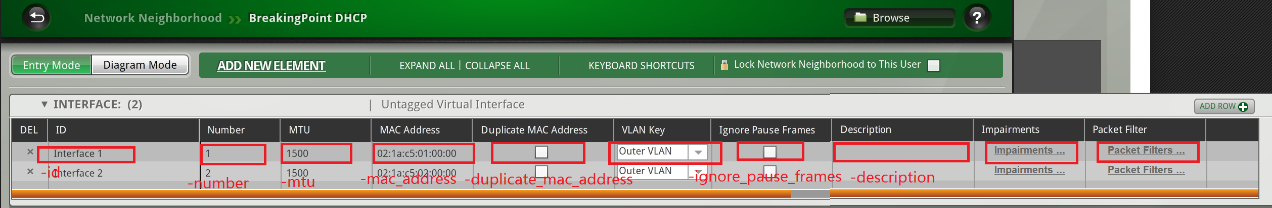
其它Element与此类同。

* args：
  + action取add时的可选参数：

针对不同Element，可以通过listElementOptions来列举可用参数：

* + - 举例interface如下：listElementOptions $networkName interface interfaceId

{-description {} {New interface}} {-duplicate\_mac\_address {} {}} {-id {} NewInterface1} {-ignore\_pause\_frames false false} {-impairments.corrupt\_chksum false false} {-impairments.corrupt\_gt256 false false} {-impairments.corrupt\_lt256 false false} {-impairments.corrupt\_lt64 false false} {-impairments.corrupt\_rand false false} {-impairments.drop false false} {-impairments.frack false false} {-impairments.rate 10 10} {-mac\_address {} 02:1A:D5:01:00:00} {-mtu 1500 1500} {-number {} 1} {-packet\_filter.dest\_ip {} {}} {-packet\_filter.dest\_port {} {}} {-packet\_filter.filter {} {}} {-packet\_filter.not\_dest\_ip false false} {-packet\_filter.not\_dest\_port false false} {-packet\_filter.not\_src\_ip false false} {-packet\_filter.not\_src\_port false false} {-packet\_filter.not\_vlan false false} {-packet\_filter.src\_ip {} {}} {-packet\_filter.src\_port {} {}} {-packet\_filter.vlan {} {}} {-vlan\_key outer\_vlan outer\_vlan}

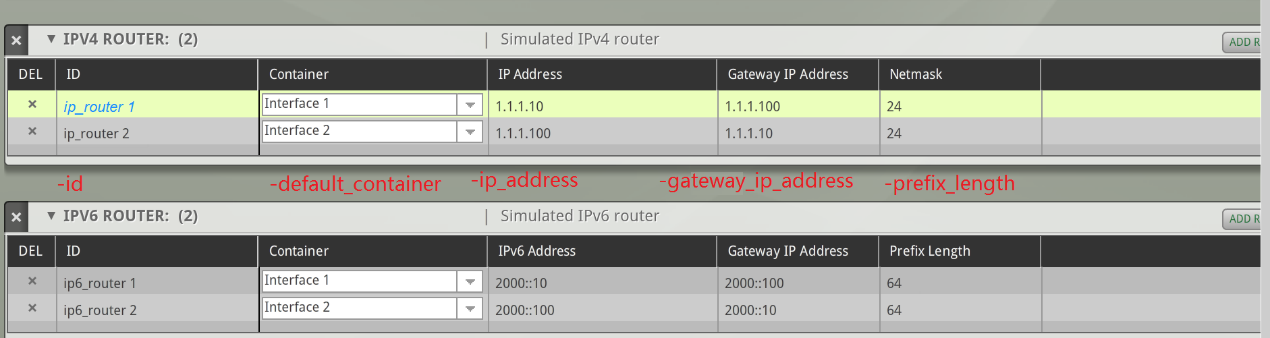


* + - 举例ip\_router如下：listElementOptions $networkName ip\_router {“routerId 1”}

{-default\_container {} {Interface 1}} {-gateway\_ip\_address {} 1.1.1.100} {-id {} {ip\_router 1}} {-ip\_address {} 1.1.1.10} {-netmask {} 24}

* + - 举例ip6\_router如下：listElementOptions $networkName ip6\_router {“routerId 1”}

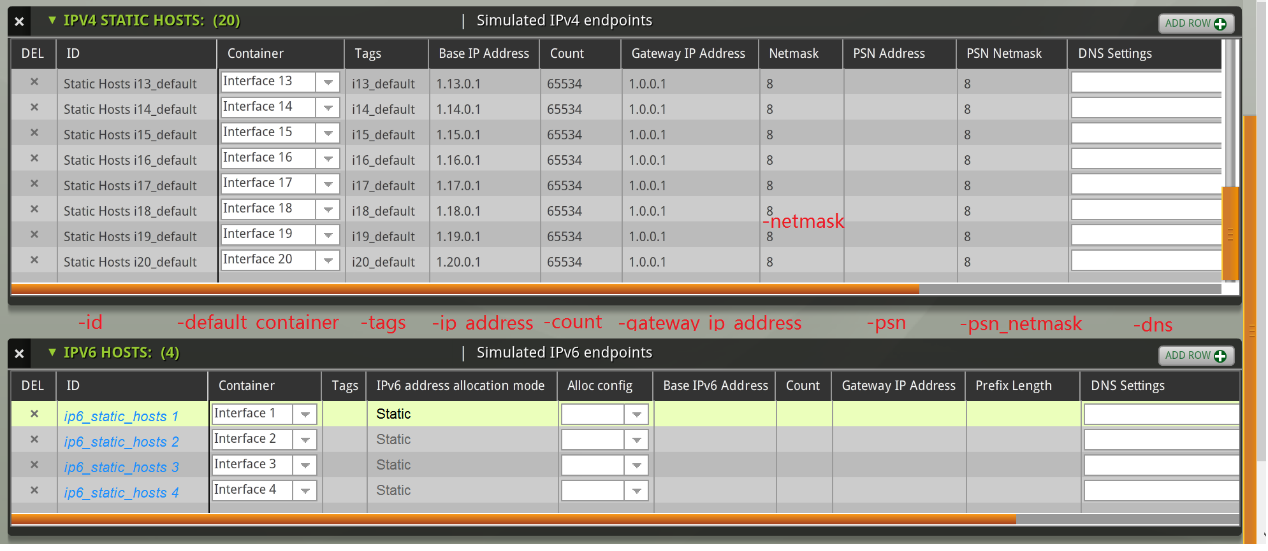
{-default\_container {} {Interface 1}} {-gateway\_ip\_address {} 2000::100} {-id {} {ip6\_router 1}} {-ip\_address {} 2000::10} {-prefix\_length {} 64}



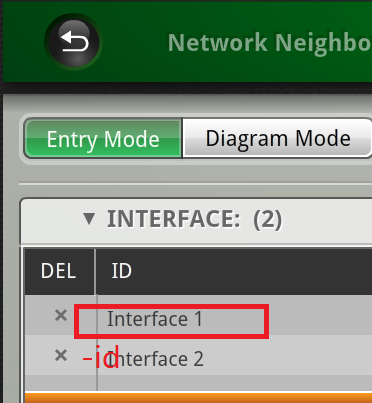
* + - 举例ip\_static\_hosts如下：listElementOptions $networkName ip\_static\_hosts {“Static Hosts i1\_default”}

{-behind\_snapt false false} {-count 1 65534} {-default\_container {} {Interface 13}} {-dns {} {}} {-enable\_stats false false} {-gateway\_ip\_address {} 1.0.0.1} {-id {} {Static Hosts i13\_default}} {-ip\_address {} 1.13.0.1} {-netmask {} 8} {-psn {} {}} {-psn\_netmask 8 8} {-tags {} i13\_default}

* + - 举例ip6\_static\_hosts如下：listElementOptions $networkName ip6\_static\_hosts {“Static Hosts i1\_default”}



* + action取config时的可选参数和add相同
  + action取remove时的唯一参数是ElementId：



### 示例

|  |
| --- |
|  |

**@tester createNetwork $newNetwork -template $network**

**set profile [ @tester getNetwork $newNetwork ]**

**#====================== Network ===========================**

**set action "add"**

**set type "interface"**

**set parameters [ list ]**

**@tester configureNetwork $newNetwork $action $type $parameters \**

**-description "New interface" \**

**-mac\_address "02:1A:D5:01:00:00" \**

**-number 1 \**

**-id NewInterface1**

**set action "config"**

**set type "interface"**

**set parameters [ list NewInterface1]**

**@tester configureNetwork $newNetwork $action $type $parameters -description "Updated interface"**

**set action "remove"**

**set type "interface"**

**set parameters [ list NewInterface1 ]**

**@tester configureNetwork $newNetwork $action $type $parameters**

**set action "add"**

**set type "path"**

**set parameters [ list path1 path2 ]**

**@tester configureNetwork $newNetwork $action $type $parameters**

**set action "remove"**

**set type "path"**

**set parameters [ list path1 path2 ]**

**@tester configureNetwork $newNetwork $action $type $parameters**

## 配置Component方法：configureComponent

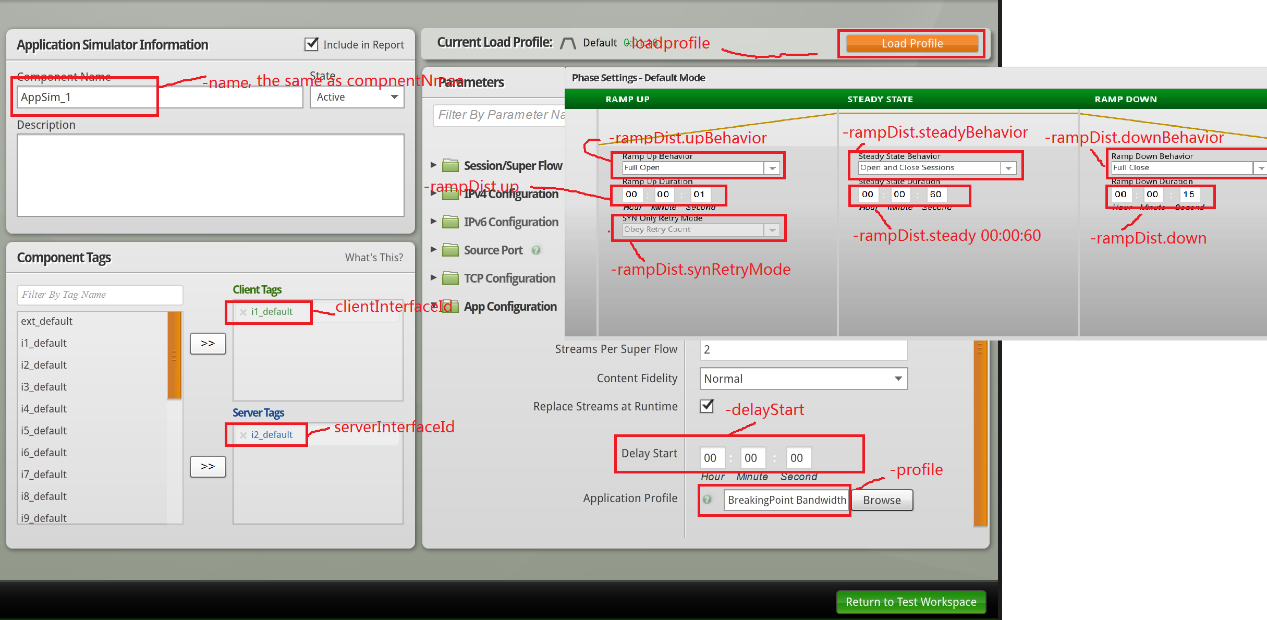
|  |
| --- |
|  |

**@tester configureComponent testName componentName args**

### 参数

|  |
| --- |
|  |

* testName：Test名字
* componentName：Component名字
* args：可选参数很多，列举如下：



{-app.fidelity {} {}} {-app.replace\_streams {} {}} {-app.streamsPerSuperflow 2 2} {-client\_tags i1\_default i1\_default} {-delayStart 00:00:00 00:00:00} {-description {A reasonable starting-point for a custom test。}} {-ip.tos 0 0} {-ip.ttl 32 32} {-ip6.flowlabel 0 0} {-ip6.hop\_limit 64 64} {-ip6.traffic\_class 0 0} {-loadprofile None None} {-name MyAutoComponent MyAutoComponent} {-profile {BreakingPoint Bandwidth} {BreakingPoint Bandwidth}} {-rampDist.down 00:00:15 00:00:15} {-rampDist.downBehavior full full} {-rampDist.steady 00:00:60 00:00:60} {-rampDist.steadyBehavior cycle cycle} {-rampDist.synRetryMode obey\_retry obey\_retry} {-rampDist.up 00:00:01 00:00:01} {-rampDist.upBehavior full full} {-rampUpProfile.increment {} {}} {-rampUpProfile.interval {} {}} {-rampUpProfile.max {} {}} {-rampUpProfile.min {} {}} {-rampUpProfile.type calculated calculated} {-rateDist.max {} {}} {-rateDist.min 10000 10000} {-rateDist.scope aggregate aggregate} {-rateDist.type constant constant} {-rateDist.unit mbps mbps} {-rateDist.unlimited false false} {-server\_tags i2\_default i2\_default} {-sessions.allocationOverride {} {}} {-sessions.closeFast true true} {-sessions.emphasis {} {}} {-sessions.engine {} {}} {-sessions.max 100000 100000} {-sessions.maxActive 0 0} {-sessions.maxPerSecond 100000 100000} {-sessions.openFast false false} {-sessions.statDetail {} {}} {-sessions.target 1 1} {-sessions.targetMatches {} {}} {-sessions.targetPerSecond 1 1} {-srcPortDist.max 65535 65535} {-srcPortDist.min 1024 1024} {-srcPortDist.type random random} {-tcp.add\_timestamps true true} {-tcp.aging\_time 0 0} {-tcp.aging\_time\_data\_type {} {}} {-tcp.delay\_acks true true} {-tcp.ecn {} {}} {-tcp.handshake\_data false false} {-tcp.initial\_congestion\_window {} {}} {-tcp.initial\_receive\_window 5792 5792} {-tcp.mss 1460 1460} {-tcp.raw\_flags {} {}} {-tcp.reset\_at\_end false false} {-tcp.retries 3 3} {-tcp.retry\_quantum\_ms 500 500} {-tcp.shutdown\_data {} {}} {-tcp.tcp\_connect\_delay\_ms {} {}} {-tcp.tcp\_keepalive\_timer {} {}} {-tcp.tcp\_window\_scale {} {}}

### 示例

|  |
| --- |
|  |

**@tester configure $testName $componentName -rampDist.up 00:00:10 \**

**-rampDist.upBehavior full \**

**-rampDist.steady 00:00:60 \**

**-rampDist.steadyBehavior cycle \**

**-rampDist.down 00:00:05 \**

**-rampDist.downBehavior full \**

**-delayStart 00:00:05 \**

**-loadprofile $newLoadProfileName**

## 配置Superflow方法：configureSuperflow

|  |
| --- |
|  |

**@tester configureSuperflow name action type parameters args**

### 参数

|  |
| --- |
|  |

* name：Superflow名字
* action：可以是add：添加一个superflow，modify：配置一个superflow， unset：unset superflow的某个属性和remove：删除一个superflow
* type：可用的Superflow资源，如：flow，action，host和maction
* parameters：根据不同Superflow资源，给予不同的参数列表：
  + type为flow：
    - action为add， parameters为[ list protocol from to ]三元组列表，例：[ list httpadv Client Server ]
    - action为modify，parameters为[ list actionid ]，例：[ list 2 ]
    - action为unset，parameters为[ list actionid ]，例：[ list 2 ]
    - action为remove，parameters为[ list actionid ]，例：[ list 2 ]
  + type为action：
    - action为add， parameters为[ list flowid source type ]三元组列表，例：[ list 1 client get\_uri ]
    - action为modify，parameters为[ list actionid ]，例：[ list 2 ]
    - action为unset，parameters为[ list actionid ]，例：[ list 2 ]
    - action为remove，parameters为[ list actionid ]，例：[ list 2 ]
  + type为host：
    - action为add， parameters为[ list name iface dnsname ]三元组列表，例：[ list HttpServer target http%n ]
    - action为modify，parameters为[ list name ]，例：[ list HttpServer ]
    - action为remove，parameters为[ list name]，例：[ list HttpServer ]
  + type为maction：
    - action为add， parameters为[ list actionid match flowid source name ]三元组列表，例：[ list 5 1 1 client get\_uri ]
    - action为modify，parameters为[ list actionid matchid subid ]，例：[ list 5 1 1 ]
    - action为unset，parameters为[ list actionid matchid subid ]，例：[ list 5 1 1 ]
    - action为remove，parameters为[ list actionid matchid subid ]，例：[ list 5 1 1 ]
* args：根据type会取不同的选项：
  + type是action：

1 {source client flowid 1 type get\_uri proxied false method GET uri {} keep-alive true accept {} accept-encoding {} accept-language {} user-agent {} cookie-name {} cookie-value {} custom-header-name {} custom-header-value {} custom\_header\_behavior replace} 2 {source server flowid 1 type response\_ok compression none keep-alive true content-md5 false chunked false content-type {} configuration\_file generate\_attachment.json response-data {} html\_inline\_text\_language English exclude\_etag false cookie-name {} cookie-value {} custom-header-name {} custom-header-value {} custom\_header\_behavior replace} 3 {source server flowid 1 type tls\_accept tls\_enabled true tls\_min\_version TLS\_VERSION\_3\_0 tls\_max\_version TLS\_VERSION\_3\_0 tls\_ciphers TLS\_CIPHERSUITE\_ALL tls\_ciphers2 TLS\_CIPHERSUITE\_NONE tls\_ciphers3 TLS\_CIPHERSUITE\_NONE tls\_ciphers4 TLS\_CIPHERSUITE\_NONE tls\_ciphers5 TLS\_CIPHERSUITE\_NONE tls\_resume\_max 100 tls\_resume\_expire 300 tls\_handshake\_timeout 0 tls\_client\_auth\_enabled false tls\_own\_cert BreakingPoint\_serverA\_512.crt tls\_own\_key BreakingPoint\_serverA\_512.key tls\_own\_dh\_params BreakingPoint\_server\_dhparams\_128.pem tls\_peer\_common\_name clientA\_512.client.int tls\_peer\_ca\_cert BreakingPoint\_cacert\_client.crt tls\_peer\_cert\_verify\_mode TLS\_CERT\_VERIFY\_NOCHECK tls\_decrypt\_mode L4\_TLS\_DECRYPT\_MODE\_AUTO} 4 {source server flowid 1 type tls\_accept tls\_enabled true tls\_min\_version TLS\_VERSION\_3\_0 tls\_max\_version TLS\_VERSION\_3\_0 tls\_ciphers TLS\_CIPHERSUITE\_ALL tls\_ciphers2 TLS\_CIPHERSUITE\_NONE tls\_ciphers3 TLS\_CIPHERSUITE\_NONE tls\_ciphers4 TLS\_CIPHERSUITE\_NONE tls\_ciphers5 TLS\_CIPHERSUITE\_NONE tls\_resume\_max 100 tls\_resume\_expire 300 tls\_handshake\_timeout 0 tls\_client\_auth\_enabled false tls\_own\_cert BreakingPoint\_serverA\_512.crt tls\_own\_key BreakingPoint\_serverA\_512.key tls\_own\_dh\_params BreakingPoint\_server\_dhparams\_128.pem tls\_peer\_common\_name clientA\_512.client.int tls\_peer\_ca\_cert BreakingPoint\_cacert\_client.crt tls\_peer\_cert\_verify\_mode TLS\_CERT\_VERIFY\_NOCHECK tls\_decrypt\_mode L4\_TLS\_DECRYPT\_MODE\_AUTO}

* + type是host：

Client {iface origin dnsname client%n}

Server {iface target dnsname server%n}

* + type是maction：

-transflag continue -proxied false -method GET -uri {} -http\_version\_override 1.1 -uri\_escape true -keep-alive true -accept {} -accept-encoding {} -accept-language {} -user-agent {} -if-none-match {} -cookie-name {} -cookie-value {} -custom-header-name {} -custom-header-value {} -custom-headers-file {} -custom\_header\_behavior replace

* + type是flow：

1 {protocol httpadv from Client to Server flow\_mode sync l4transport 0 sctp-port 9899 sctp-checksum-type CRC32 sctp-shared-flow-enabled 0 sctp-key-offset-client 12 sctp-key-offset-server 12 sctp-shared-flow-master-only-enabled 0 ip4\_tos\_dscp 0 ip6\_traffic\_class 0 ip6\_flowlabel 0 mobile\_bearer\_id 5 server-port-range-full Disabled client-profile desktop\_mix server-profile default http\_version 1.1 server-hostname default persist-cookies true num-rand-cookies 0 min-cookie-length 1 max-cookie-length 15 rand-cookie-value-persist false client-port 0 server-port 80}

### 示例

|  |
| --- |
|  |

**@tester createSuperflow $newSuperflow -template $superflow**

**set flow [ @tester getSuperflow $newSuperflow ]**

**#====================== Flow ===========================**

**set type "flow"**

**set action "add"**

**#set parameters [ list protocol from to ]**

**set parameters [ list httpadv Client Server ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters**

**set action "modify"**

**#set parameters [ list actionid ]**

**set parameters [ list 2 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -to Client -from Server -client-profile ie**

**set action "unset"**

**set parameters [ list 2 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -client-profile**

**set action "remove"**

**#set parameters [ list actionid ]**

**set parameters [ list 2 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters**

**#====================== Flow ===========================**

**#====================== Action ===========================**

**set type "action"**

**set action "add"**

**#set parameters [ list flowid source type ]**

**set parameters [ list 1 client get\_uri ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -proxied true**

**set action "modify"**

**#set parameters [ list actionid ]**

**set parameters [ list 3 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -proxied true**

**set action "unset"**

**set parameters [ list 3 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -proxied**

**set parameters [ list 4 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -proxied**

**set action "remove"**

**#set parameters [ list actionid ]**

**set parameters [ list 4 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters**

**#====================== Action ===========================**

**#====================== Host ===========================**

**set type "host"**

**set action "add"**

**#set parameters [ list name iface dnsname ]**

**set parameters [ list HttpServer target http%n ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters**

**set action "modify"**

**#set parameters [ list name ]**

**set parameters [ list HttpServer ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -iface origin**

**set action "remove"**

**#set parameters [ list name ]**

**set parameters [ list HttpServer ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters**

**#====================== Host ===========================**

**#====================== Mathch Action ===========================**

**set type "action"**

**set action "add"**

**#set parameters [ list flowid source type ]**

**set parameters [ list 1 client expect ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters \**

**-match1 {200 OK} \**

**-match2 {301 Moved} \**

**-match3 {404 Not} \**

**-nomatch.timeout 3**

**set type "maction"**

**set action "add"**

**#set parameters [ list actionid match flowid source name ]**

**set parameters [ list 5 1 1 client get\_uri ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -uri /match1.html**

**set action "modify"**

**#set parameters [ list actionid matchid subid ]**

**set parameters [ list 5 1 1 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -uri /match2.html**

**set action "unset"**

**#set parameters [ list actionid matchid subid ]**

**set parameters [ list 5 1 1 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters -uri**

**set action "remove"**

**#set parameters [ list actionid matchid subid ]**

**set parameters [ list 5 1 1 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters**

**set type "action"**

**set action "remove"**

**#set parameters [ list actionid ]**

**set parameters [ list 4 ]**

**@tester configureSuperflow $newSuperflow $action $type $parameters**

**#====================== Action ===========================**

## 配置AppProfile方法：configureAppProfile

|  |
| --- |
|  |

**@tester configureAppProfile appProfileName superFlowName action args**

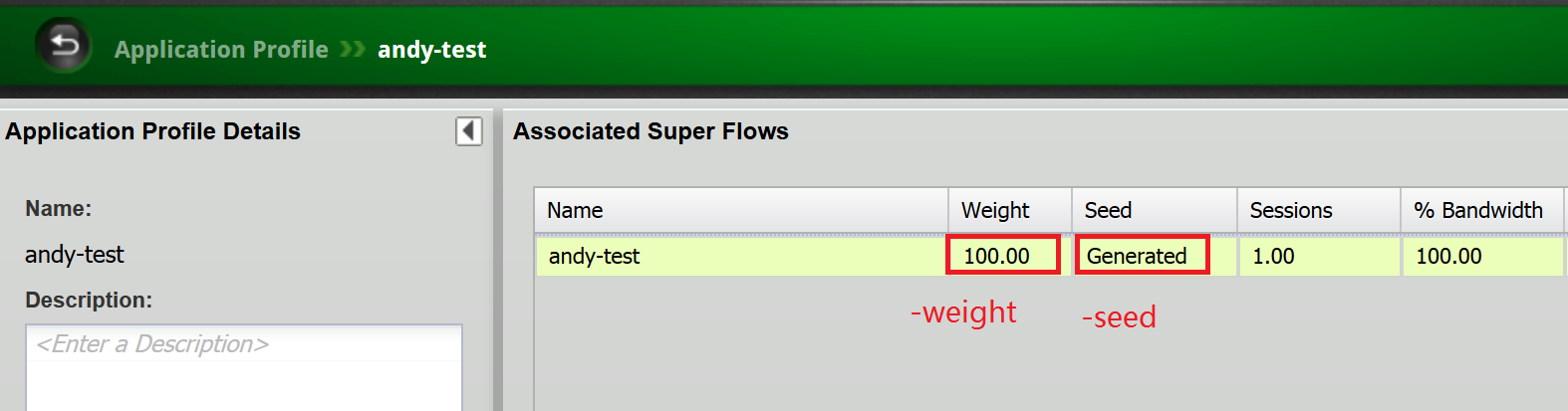
### 参数

|  |
| --- |
|  |

* appProfileName：AppProfile名字
* superFlowName：Superflow名字
* action：可以是add：添加一个superflow，modify：配置一个superflow和remove：删除一个superflow
* args：可选参数如下：

-weight：flow在AppProfile中所占的权重

-seed：随机数策略值



### 示例

|  |
| --- |
|  |

**@tester createAppProfile $newAppProfile -template $appProfile**

**set profile [ @tester getAppProfile $newAppProfile ]**

**#====================== Flow ===========================**

**set action "add"**

**@tester configureAppProfile $newAppProfile $superflow $action -weight 1111**

**set action "modify"**

**@tester configureAppProfile $newAppProfile $superflow $action -weight 888 -seed 999**

**set action "remove"**

**@tester configureAppProfile $newAppProfile $superflow $action**

**#====================== Flow ===========================**

# 导入导出资源

## 导入Test：importTest

|  |
| --- |
|  |

**@tester importTest name args**

### 参数

|  |
| --- |
|  |

* name：重命名导入Test名字
* args：对应参数描述如下：

-file： 本地要导入Test文件的绝对路径

-force： 次flag表明强制覆盖BPS系统已有Test

### 示例

|  |
| --- |
|  |

**@tester importTest Import1 -file C:/Tmp/Bps/import.bpt –force**

## 导出Test：exportTest

|  |
| --- |
|  |

**@tester exportTest name args**

### 参数

|  |
| --- |
|  |

* name：导出Test名字 ，如果缺省name，那么必须指定-testid value
* args：对应参数描述如下：

-file：导出Test到本地文件的绝对路径

-force：次flag表明强制覆盖BPS系统已有Test

-testid：如果缺省name，请加上次option

### 示例

|  |
| --- |
|  |

**@tester exportTest export1 -file C:/Tmp/Bps/import.bpt –force**

**Or**

**@tester exportTest –testid** [**test@64@127.0.0.1@DSTA\_Test4\_Compromise\_Microsoft\_ph2@7**](mailto:test@64@127.0.0.1@DSTA_Test4_Compromise_Microsoft_ph2@7) **-file C:/Tmp/Bps/import.bpt –force**

## 导出Report：exportReport

|  |
| --- |
|  |

**@tester exportReport name args**

### 参数

|  |
| --- |
|  |

* name：导出Report对应的Test名字，同一个测试运行多次，默认取最新的Report
* args：对应参数描述如下：

-file: The destination file to export report

-channel: File channel to save the report, not support now

-format: Report format: csv, flowstats, html, pdf, rtf, xls, bpt, xml, zip, default value is pdf

-sectionids: Filter desired sections

-iterations: Run number or testId in form of test@<testid>@

### 示例

|  |
| --- |
|  |

**@tester exportReport MyAutoTest -file C:/Tmp/Bps/MyAutoTest\_Report.pdf –format pdf**

# 端口操作

## 配置端口方法：configurePort

|  |
| --- |
|  |

**@tester configurePort slot port args**

### 参数

|  |
| --- |
|  |

* slot：板卡号，0 base
* port：端口号，0 base
* args：可选参数如下：

-auto: Auto negotiate port settings, default value is false

-speed: Port speed, default value is 1000

-fullduplex: Wether working with full duplex mode, default value is true

-port\_force\_reserve: Force to reserve port

-port\_mtu: Specifies the default MTU

### 示例

|  |
| --- |
|  |

**@tester configurePort 1 1 –auto true –speed 100 –fullduplex true -port\_force\_reserve true**

## 占用端口方法：reservePort

|  |
| --- |
|  |

**@tester reservePort portlist args**

### 参数

|  |
| --- |
|  |

* portlist：要占用端口list，格式为：[list { 0 0 } { 0 1 }]
* args：可选参数如下：

-auto: Auto negotiate port settings, default value is false

-group: Group ID, default value is 1

### 示例

|  |
| --- |
|  |

**@tester reservePort [list { 0 0 } { 0 1 }] –group 1**

## 释放端口方法：unreservePort

|  |
| --- |
|  |

**@tester unreservePort portlist args**

### 参数

|  |
| --- |
|  |

* portlist：要占用端口list，格式为：[list { 0 0 } { 0 1 }]

### 示例

|  |
| --- |
|  |

**@tester unreservePort [list { 0 0 } { 0 1 }]**

# Capture操作

## 启动抓包：startCapture

|  |
| --- |
|  |

**@tester startCapture**

### 参数

|  |
| --- |
|  |

### 示例

|  |
| --- |
|  |

**@tester startCapture**

## 停止抓包：stopCapture

|  |
| --- |
|  |

**@tester stopCapture**

### 参数

|  |
| --- |
|  |

### 示例

|  |
| --- |
|  |

**@tester stopCapture**

## 导出抓包文件：exportCapture

|  |
| --- |
|  |

**@tester exportCapture portlist dir direction args**

### 参数

|  |
| --- |
|  |

* Portlist：要导出报文的端口列表，例：[list {1 0} {1 1}]
* dir：导出报文存放的文件夹
* direction：导出报文类型， 例：rx，tx或both
* args：例举部分可选参数如下，更多详细信息请参阅手册1152页：

-compress: Whether compress the capture data

-txsnaplen: Truncates transmitted packets that are larger than txsnaplen bytes

-rxsnaplen: Truncates received packets that are larger than rxsnaplen bytes

-txfilter: Filter packets from specified host, eg. "host 10.1.0.254"

-rxfilter: Filter packets from specified host, eg. "host 10.1.0.254"

-progress: The script to run with progress notifications

### 示例

|  |
| --- |
|  |

**@tester exportCapture [list { 0 0 } { 0 1 }] C:/Bps both –compress true**

# Stats操作

## 获取Run-Time数据：getRtStats

|  |
| --- |
|  |

**@tester getRtStats name args**

### 参数

|  |
| --- |
|  |

* name：Test 名字
* args：可选参数如下：

-filters：返回数据filter

### 示例

|  |
| --- |
|  |

**@tester getRtStats $testName**

### 返回值

|  |
| --- |
|  |

**成功返回以下数组：**

[ { tcpClientClosed: 5672,

tcpServerCloseRate: 0,

tcpClientEstablishRate: 0,

appAttempted: 18465,

ethRxFrameRate: 68.69,

tcpAvgCloseTime: 0.107,

tcpAvgResponseTime: 0.243,

ethTxFrameRate: 68.69,

t: 144.463,

progress: 100,

appSuccessfulRate: 0,

tcpAvgSetupTime: 0.31,

tcpClientEstablished: 5676,

tcpAvgSessionDuration: 273.14,

time: 147.59636,

udpFlowsConcurrent: 4,

tcpServerClosed: 5672,

tcpClientCloseRate: 0,

sctpFlowsConcurrent: 0,

tcpServerEstablishRate: 0,

appSuccessful: 18463,

ethRxFrames: 236100,

tcpAttemptRate: 0,

ethTxFrames: 236100,

ethRxFrameDataRate: 0.1365,

appAttemptedRate: 0,

tcpFlowsConcurrent: 4,

superFlowsConcurrent: 2,

tcpServerEstablished: 5676,

ethTxFrameDataRate: 0.1365,

tcpAttempted: 5676 } ]

失败返回空list：

[]

## 获取Component数据：getComponentStats

|  |
| --- |
|  |

**@tester getComponentStats name args**

### 参数

|  |
| --- |
|  |

* name：Test 名字
* args：可选参数如下：

-filters：返回数据filter

### 示例

|  |
| --- |
|  |

**@tester getComponentStats $componentName**

### 返回值

|  |
| --- |
|  |

**成功返回以下数组：**

[ { tcpClientClosed: 5672,

tcpServerCloseRate: 0,

tcpClientEstablishRate: 0,

appAttempted: 18465,

ethRxFrameRate: 68.69,

tcpAvgCloseTime: 0.107,

tcpAvgResponseTime: 0.243,

ethTxFrameRate: 68.69,

t: 144.463,

progress: 100,

appSuccessfulRate: 0,

tcpAvgSetupTime: 0.31,

tcpClientEstablished: 5676,

tcpAvgSessionDuration: 273.14,

time: 147.59636,

udpFlowsConcurrent: 4,

tcpServerClosed: 5672,

tcpClientCloseRate: 0,

sctpFlowsConcurrent: 0,

tcpServerEstablishRate: 0,

appSuccessful: 18463,

ethRxFrames: 236100,

tcpAttemptRate: 0,

ethTxFrames: 236100,

ethRxFrameDataRate: 0.1365,

appAttemptedRate: 0,

tcpFlowsConcurrent: 4,

superFlowsConcurrent: 2,

tcpServerEstablished: 5676,

ethTxFrameDataRate: 0.1365,

tcpAttempted: 5676 } ]

失败返回空list：

[]

## 获取Aggregated数据：getAggStats

|  |
| --- |
|  |

**@tester getAggStats name args**

### 参数

|  |
| --- |
|  |

* name：Test 名字
* args：可选参数如下：

-filters：返回数据filter

### 示例

|  |
| --- |
|  |

**@tester getAggStats $testName**

### 返回值

|  |
| --- |
|  |

**成功返回以下数组：**

[ {cpu\_usage: 0, #CPU Usage

ethAlignmentErrors: 0， #Ethernet alignment errors

ethDropEvents: 0, #Ethernet drop events

ethFCSErrors: 0, #Ethernet FCS errors

ethOversizedFrames: 0, #Ethernet oversize frames

ethRxErrors: 0, #Ethernet receive errors

ethRxFrameData: 0, #Ethernet bytes received. This includes L7

and all packet overhead, including L2,

L3, L4 headers, ethernet CRC, and interpacket

gap (20 bytes per frame).

ethRxFrameDataRate: 0, #Ethernet receive rate. This includes L7

and all packet overhead, including L2,

L3, L4 headers, ethernet CRC, and interpacket

gap (20 bytes per frame)

ethRxFrameRate: 0, #Ethernet frame receive rate

ethRxFrames: 0, #Ethernet frames received

ethRxPauseFrames: 0, #Ethernet pause frames received

ethTotalErrors: 0, #Total Errors

ethTxErrors: 0, #Ethernet transmit errors

ethTxFrameData: 0, #Ethernet bytes transmit. This includes L7

and all packet overhead, including L2,

L3, L4 headers, ethernet CRC, and interpacket

gap (20 bytes per frame).

ethTxFrameDataRate: 0, #Ethernet transmit rate. This includes L7

and all packet overhead, including L2,

L3, L4 headers, ethernet CRC, and interpacket

gap (20 bytes per frame).

ethTxFrameRate: 0, #Ethernet frame transmit rate

ethTxFrames: 0, #Ethernet frames transmitted

ethTxPauseFrames: 0, #Ethernet pause frames transmitted

ethUndersizedFrames: 0, #Ethernet undersize frames

linux mem\_free\_kb: 0, #Free memory on the System Controller

mem\_total\_kb: 0, #Total memory on the System Controller

mem\_used\_kb: 0, #Used memory

mount percent\_used: 0, #The percent of disk spaced used on the disk partition

superFlowRate: 0, #Super Flow rate

superFlows: 0, #Aggregate Super Flows

superFlowsConcurrent: 0, #Concurrent Super Flows

tcpFlowRate: 0, #TCP Flow rate

tcpFlows: 0, #Aggregate TCP Flows

tcpFlowsConcurrent: 0, #Concurrent TCP Flows

timestamp: 0, #The time that the datapoint was taken

(refers to the rest of the data that comes

with it)

udpFlowRate: 0, #UDP Flow rate

udpFlows: 0, #Aggregate UDP Flows

udpFlowsConcurrent: 0, #Concurrent UDP Flows} ]

失败返回空list：

[]

## 获取Process数据：getRtProcess

Test运行在同步模式下的测试进度

|  |
| --- |
|  |

**@tester getRtProcess**

### 参数

|  |
| --- |
|  |

* name：Test 名字
* args：可选参数如下：

-filters：返回数据filter

### 示例

|  |
| --- |
|  |

**@tester getRtProcess $testName**

### 返回值

|  |
| --- |
|  |

成功返回以下数组：

{ name: test1, processPercentage: 20 }

失败返回空list：

[]

# 运行、取消测试

## 运行测试：run

|  |
| --- |
|  |

**@tester run name args**

### 参数

|  |
| --- |
|  |

* name：Test 名字
* args：可选参数如下：

-allowMalware：Confirm that malware should be allowed in this test

-async: Runs the test in the background and executes the command specified

-flowexceptions: Identifies the script to run with flow exception notifications

-group: Identifies the interface group to be used

-progress: Allows you to monitor the progress of the test

-rtstats: Calls the -rtstats attribute when there are new Real-Time statistics available

### 示例

|  |
| --- |
|  |

**@tester run $testName –rtstats cbRunTimeStats**

### 返回值

|  |
| --- |
|  |

成功返回true,

失败返回false

## 取消测试：cancel

|  |
| --- |
|  |

**@tester cancel name args**

### 参数

|  |
| --- |
|  |

* name：Test 名字
* args：可选参数如下：

-async: Cancel the test in the background and executes the command specified

### 示例

|  |
| --- |
|  |

**@tester cancel $testName -async cbCancelTest**

### 返回值

|  |
| --- |
|  |

成功返回true,

失败返回false

# 保存对象

## 保存对象：save

|  |
| --- |
|  |

**@tester save name args**

### 参数

|  |
| --- |
|  |

* name：对象名字，对象可以是：Test，Network，Component，AppProfile，LoadProfile或TestSeries
* args：可选参数如下：

-name：保存对象的名字

-force：强制覆盖系统已有同名对象

### 示例

|  |
| --- |
|  |

**@tester save $objName –force**

### 返回值

|  |
| --- |
|  |

成功返回true,

失败返回false

# 删除对象

## 删除对象：delete

|  |
| --- |
|  |

**@tester delete name args**

### 参数

|  |
| --- |
|  |

* name：对象名字，对象可以是：Test，Network，Component，AppProfile，LoadProfile或TestSeries
* args：可选参数如下：

-force：强制删除对象

### 示例

|  |
| --- |
|  |

**@tester delete $objName –force**

### 返回值

|  |
| --- |
|  |

成功返回true,

失败返回false

# 获取对象方法

## 获取对象：getObjByName

|  |
| --- |
|  |

**@tester getObjByName name**

**通过指定的name，获取Test，TestSeries，Neigborhood，LoadProfile，Component，AppProfile及StrikeList对象**

### 参数

|  |
| --- |
|  |

* name：对象名字，对象可以是：Test，Neighborhood，Component，AppProfile，LoadProfile，StrikeList或TestSeries

### 示例

|  |
| --- |
|  |

**@tester getObjByName $objName**

### 返回值

|  |
| --- |
|  |

成功返回：对象句柄

失败返回：””

# 备注

1. 如果使用到参数， 如对象到名字等携带空格字符，请在传递过程中最好进行如下处理：

listElementOptions $networkName interface {“abc xyz”}

或

listElementOptions $networkName interface “{abc xyz}”

1. 用如下方法查看对象可配置参数和目前对象属性值：

$object configure

1. 用如下方法查看对象可用方法：

$object