function.tcl 单元测试

测试版本：release version 1.8

1. StartQT

函数功能：程序主函数，用于判断输入参数合法性；配置、运行并保存测试结果。

函数设计者：Tim

测试环境：IxNetwork, Wish Console for IxNetwork,

测试类型：单元测试

1. StartQT传入异常参数

测试目的：StartQT传入异常参数，是否报错

测试过程：打开IxNetwork及Wish Console for IxNetwork，导入function.tcl

1. 测试输入：StartQT -port\_list 192.168.200.202/1/1

预期输出：ERROR: -port\_list format is not right, the value is 192.168.200.202/1/1

实际输出：同预期输出

1. 测试输入：StartQT -latency\_type cut

预期输出：ERROR: -latency\_type format is not right, the value is cut, should be CutThrough|StoreAndForward

实际输出：同预期输出

1. 测试输入：StartQT -hol\_type one

预期输出：ERROR: -hol\_type format is not right, the value is one, should be oneGroup|manyGroup

实际输出：同预期输出

1. 测试输入：StartQT -jumbo\_value 9356.2

预期输出：Error:jumbo\_value format error:jumbo\_value: 9356.2 ,should be integer

实际输出：同预期输出

1. 测试输入：StartQT -test\_duration 25s

预期输出：Error:test\_duration format error:test\_duration:25s ,should be integer

实际输出：同预期输出

1. 测试输入：StartQT -lng\_rate 12.5

预期输出：Error:lng\_rate format error:lng\_rate:12.5

实际输出：同预期输出

1. 测试输入：StartQT -frames\_per\_addr 3a

预期输出：Error:frames\_per\_addr format error:frames\_per\_addr:3a

实际输出：同预期输出

1. 测试输入：StartQT -media pos

预期输出：Error:media format error: media: pos , should be fiber | copper

实际输出：同预期输出

1. 测试输入：StartQT -autoneg auto

预期输出：Error:autoneg format error: autoneg: auto , should be Auto | Half | Full

实际输出：同预期输出

1. 测试输入：StartQT -protocol IPv4 -ip\_addr\_start 2002::1

预期输出：Error:ipv4 addrss format error: ip\_addr\_start: 2002::1 ; ip\_addr\_step: ; gw\_addr\_start: ;gw\_addr\_step: ;port\_step:

实际输出：同预期输出

1. 测试输入：StartQT -protocol IPv6 -ip\_addr\_step 0.0.0.1

预期输出：Error:ipv6 addrss format error: ip\_addr\_start: ;ip\_addr\_step: 0.0.0.1;

gw\_addr\_start: ; gw\_addr\_step: ; port\_step:

实际输出：同预期输出

1. 测试输入：StartQT -protocol IPv4 -pfx\_len 48

预期输出：Error:ipv4 pfx\_len error: pfx\_len: 48

实际输出：同预期输出

1. 测试输入：StartQT -protocol IPv6 -pfx\_len 64.5

预期输出：Error:ipv6 pfx\_len error: pfx\_len: 64.5

实际输出：同预期输出

1. 测试输入：StartQT -quicktest “”

预期输出：Error:No quickTest is choosed

实际输出：同预期输出

1. StartQT 传入正常参数

测试目的：传入正常参数，检查StartQT是否正确执行，是否有结果文件生成

测试过程：打开IxNetwork及Wish Console for IxNetwork，导入function.tcl

测试输入：StartQT -port\_list {192.168.2.202:1:5 192.168.2.202:1:6 192.168.2.202:1:7 192.168.2.202:1:8} -2544\_throughput\_enable 1 -latency\_type CutThrough -pair\_enable 1 -ip\_enable 1 -ip\_addr\_start 100.0.1.2 -ip\_addr\_step 0.0.1.0 -gw\_addr\_start 100.0.1.1 -gw\_addr\_step 0.0.1.0 -pfx\_len 24 -port\_step 0.1.0.0 -fs64\_enable 1 -test\_duration 10 -lng\_rate 20 -frames\_per\_addr 10 -send\_mac\_only\_enable 1 -1g\_enable 1 -media fiber -autoneg Auto

预期结果：在IxNetwork软件界面中新建了一个Quicktest项目，测试协议为RFC2544 Throughput，并占了IP为192.168.2.202机框下第一张卡的5-8口作为测试端口，测试配置同输入参数，并开始测试，完成测试后在默认测试结果保存路径下有测试结果文件夹生成，文件夹内有详细测试结果文件。

实际结果：同预期结果

1. StopQT

函数功能：在测试过程中停止测试

函数设计者：Tim

测试环境：IxNetwork, Wish Console for IxNetwork,

测试类型：单元测试

测试过程：打开IxNetwork及Wish Console for IxNetwork，导入function.tcl

测试输入：首先手动在IxNetwork软件内创建一个完整的Quicktest测试，并开始测试，然后在tcl解析器中输入StopQT

预期结果：正在运行中的Quicktest项目停止运行

实际结果：同预期输出

1. IsIPv4Address

函数功能：对于输入的值进行判断是否是正确格式的ipv4地址

函数设计者：Cathy

测试环境：Wish Console for IxNetwork

测试类型：单元测试

测试过程：打开Wish Console for IxNetwork，导入function.tcl

测试输入：

1. IsIPv4Address 1.0.0.0
2. IsIPv4Address 255.255.255.255
3. IsIPv4Address 256.1.1.1
4. IsIPv4Address AA.BB.CC.DD
5. IsIPv4Address 1.2.3.-4

预期输出：

1. 1
2. 1
3. 0
4. 1
5. 0

实际输出：

1. 1
2. 1
3. 0
4. 0
5. 1

问题报告：对于输入值有负数的IPv4地址，应返回0实际返回1，对于16进制格式的IPv4地址判断错误，应返回1，实际返回0，该bug将于下一版本（v1.9）中修复。

1. IsIPv6Address

函数功能：对于输入的值进行判断是否是正确格式的ipv6地址

函数设计者：Cathy

测试环境：Wish Console for IxNetwork

测试类型：单元测试

测试过程：打开Wish Console for IxNetwork，导入function.tcl

测试输入：

1. IsIPv6Address 2001:0:0:0:0:0:0:1
2. IsIPv6Address 2001::1
3. IsIPv6Address AA:BB::1
4. IsIPv6Address 2000::-1

预计输出：

1. 1
2. 1
3. 1
4. 0

实际输出：

1. 1
2. 1
3. 1
4. 1

问题报告：对于输入值有负数的IPv6地址判断错误，应返回0，实际返回1，该bug将于下一版本（v1.9）中修复。

1. IncrMacAddr

函数功能：对于输入的mac1进行增加操作，增量为mac2，默认增量为00:00:00:00:01:00

函数设计者：Cathy

测试环境：Wish Console for IxNetwork

测试类型：单元测试

测试过程：打开Wish Console for IxNetwork，导入function.tcl

测试输入：

1. IncrMacAddr 00:00:00:00:00:01
2. IncrMacAddr 00:00:00:00:00:01 00:00:00:00:00:01
3. IncrMacAddr 00:00:00:00:FF:00

预计输出：

1. 00:00:00:00:01:01
2. 00:00:00:00:00:02
3. 00:00:00:00:01:00

实际输出：同预计输出

1. GetRealPort

函数功能：检查输入的机框、板卡及端口是否存在，如果存在，检查端口是否可用，如果不可用，则强行释放该端口。

函数设计者：Tim

测试环境：IxNetwork, Wish Console for IxNetwork,

测试类型：单元测试

测试过程：打开IxNetwork及Wish Console for IxNetwork，导入function.tcl

测试输入：

1. GetRealPort 192.168.2.202:1:5
2. GetRealPort 192.168.2.202:3:1
3. 在IxNetwork界面上占用机框192.168.2.202上第一张卡第5个口，然后输入(1)中相同输入

预期输出：

1. 192.168.2.202/card:1/port:5
2. ixNet getNull
3. 192.168.2.202/card:1/port:5同时观察IxNetwork软件界面，占用的端口被释放

实际输出：同预期输出

1. QTP\_ChasInfo

函数功能：获取输入机框的信息。

函数设计者：Tim

测试环境：IxNetwork, Wish Console for IxNetwork,

测试类型：单元测试

测试过程：打开IxNetwork及Wish Console for IxNetwork，导入function.tcl

测试输入：

1. QTP\_ChasInfo 192.168.2.202
2. QTP\_ChasInfo 192.168.2.100
3. QTP\_ChasInfo w.w.w.w

预期输出：

1. out=192.168.2.202:1:1 192.168.2.202:1:2 192.168.2.202:1:3 192.168.2.202:1:4 192.168.2.202:1:5 192.168.2.202:1:6 192.168.2.202:1:7 192.168.2.202:1:8 192.168.2.202:1:9 192.168.2.202:1:10 192.168.2.202:1:11 192.168.2.202:1:12 192.168.2.202:1:13
2. <error>|msg=chassis connection timeout
3. <error>|msg=chassis connection timeout

实际输出：同预期输出

1. QTP\_PortListSet

函数功能：创建vport，并将虚拟端口与实际端口对应连接起来。

函数设计者：Tim

测试环境：IxNetwork, Wish Console for IxNetwork,

测试类型：单元测试

前置条件：完成GetRealPort测试

测试过程：打开IxNetwork及Wish Console for IxNetwork，导入function.tcl

测试输入：

1. QTP\_PostListSet –port\_list {192.168.2.202:1:5 192.168.2.202:1:6 192.168.2.202:1:7 192.168.2.202:1:8}
2. QTP\_PostListSet –port\_list {192.168.2.100:1:5 192.168.2.100:1:6 192.168.2.100:1:7 192.168.2.100:1:8}

预期输出:

1. ::ixNet::OBJ-/vport:1 ::ixNet::OBJ-/vport:2 ::ixNet::OBJ-/vport:3 ::ixNet::OBJ-/vport:4 同时打开软件界面，在port下查看4个端口对应连接的是机框192.168.2.202下第一张卡的5，6，7，8四个端口
2. The real card is not exist!

NAK

实际输出：同预期输出

1. QTP\_NewQTobj

函数功能：创建Quicktest测试

函数设计者：Tim

测试环境：IxNetwork, Wish Console for IxNetwork,

测试类型：单元测试

测试过程：打开IxNetwork及Wish Console for IxNetwork，导入function.tcl

测试输入：

1. QTP\_NewQTobj –quicktest {rfc2544throughput rfc2889fullyMeshed}
2. 第一次输入完成后不清空配置，在软件界面上修改部分参数后，再输入一次(1)中的输入内容

预期输出：

1. rfc2889fullyMeshed ::ixNet::OBJ-/quickTest/rfc2889fullyMeshed:1 rfc2544throughput ::ixNet::OBJ-/quickTest/rfc2544throughput:1在软件界面上查看新建了两个Quicktest测试
2. 原先的两个Quicktest测试（经过配置的）被清空了，两个新的Quicktest测试被创建

实际输出：同预期输出

1. AddPortocol | TrafficSet |QTP\_EndPointSet | QTP\_EndPointSet | QTP\_TrafficeSet |QTP\_GetResult

函数功能：对测试进行配置

函数设计者：Tim

测试环境：IxNetwork, Wish Console for IxNetwork,

测试类型：集成测试

测试过程：此部分函数应用于StartQT函数中，异常值测试统一在StartQT中完成测试，不会再传入异常值，正常值测试也在StartQT测试过程中一并完成，证明功能正常，因此不再单独进行单元测试。