IneoQuest Tcl API

2.20 for Tcl API 2.9.0

Generated by Doxygen at: Fri Jan 8 00:00:21 2010

Notice

The information in this guide is subject to change without notice.

INEOQUEST TECHNOLOGIES, INCORPORATED SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN; NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

The software described in this guide is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

IneoQuest Technologies, Inc., 170 Forbes Boulevard, Mansfield, Massachusetts 02048 USA

The following are trademarks of IneoQuest Technologies, Inc.:

IneoQuest Technologies, Singulus, iVMS, IQDVx, IQMediaStim, IQTsxPro, IQMediaAnalyzer Pro, Cricket, IQDialogue, IQPinPoint, IQWatch, RVL, IQtv, IQMediaMonitor, Geminus, Get the Picture, Multi-Dimensional Video Quality Monitoring, Revenue Assurance, IQVisionProbe, IQRouterTest, SmartVIEW, IQMediaMonitor100 and IQMediaSentry

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Contents

1 IneoQuest Tcl API Reference Manual **1**

1.1 General API Information 1

2 Command Line Interface for IneoQuest Tcl API **3**

2.1 Stimulus-Mode Configuration Files 5

2.2 Analysis-Mode Configuration Files 6

2.3 CLI Simulus-Mode Examples 8

2.4 CLI Analysis-Mode Examples 9

2.4.1 Check Flow Parameters 9

2.4.2 Trigger Video Captures 9

3 Tcl Documentation Revision History **11**

3.1 Revision History 12

4 Installation Procedure **15**

4.1 Installation on Windows 16

4.1.1 Setting Windows Environment Variables 16

4.2 Installation on Linux 17

4.2.1 Linux Environment Variables 17

4.3 Installation on Solaris 18

4.3.1 Setting Solaris Environment Variables 19

5 Module Documentation **21**

5.1 Alarm Methods 22

5.1.1 Detailed Description 22

5.1.1.1 iqtcl_ClearAlarms 23

5.1.1.2 iqtcl_ClearAlarmLog 24

5.1.1.3 iqtcl_GetActiveAlarmFirst 25

5.1.1.4 iqtcl_GetActiveAlarmNext 26

5.1.1.5 iqtcl_GetActiveAlarmTable 27

5.1.1.6 iqtcl_GetAlarmLogFirst 28
5.1.1.7 iqtcl_GetAlarmLogNext 29
5.1.1.8 iqtcl_GetAlarmLogTable 30
5.2 Alias Methods 31
5.2.1 Detailed Description 31
5.2.1.1 iqtcl_GetAliasFirst 32
5.2.1.2 iqtcl_GetAliasNext 33
5.2.1.3 iqtcl_ClearAliases 34
5.3 ASI Methods 35
5.3.1 Detailed Description 35
5.3.1.1 iqtcl_StartASIScan 36
5.3.1.2 iqtcl_StopASIScan 37
5.3.1.3 iqtcl_GetASIStatus 38
5.3.1.4 iqtcl_ASILockStream 39
5.3.1.5 iqtcl_ASIUnlockStream 40
5.4 Capture Methods 41
5.4.1 Detailed Description 41
5.4.1.1 iqtcl_OpenCapture 42
5.4.1.2 iqtcl_OpenStreamCapture 43
5.4.1.3 iqtcl_CloseCapture 44
5.4.1.4 iqtcl_StartCapture 45
5.4.1.5 iqtcl_StopCapture 46
5.4.1.6 iqtcl_UploadCapture 47
5.4.1.7 iqtcl_GetCaptureStatus 48
5.5 Census Methods 49
5.5.1 Detailed Description 49
5.5.1.1 iqtcl_GetCensusFirst 50
5.5.1.2 iqtcl_GetCensusNext 51
5.5.1.3 iqtcl_GetCensusByID 52
5.5.1.4 iqtcl_GetNextCensusID 53
5.5.1.5 iqtcl_GetStreamID 54
5.5.1.6 iqtcl_ClearCensus 55
5.5.1.7 iqtcl_ClearStream 56
5.5.1.8 iqtcl_ClearStreamStats 57
5.5.1.9 iqtcl_GetCensusTableItem 58
5.5.1.10 iqtcl_GetCensusTableItemTagged 59

5.5.1.11 iqtcl_GetQAMTableItem 60
5.5.1.12 iqtcl_GetQAMTableItemTagged 61
5.5.1.13 iqtcl_GetQAMTable 62
5.5.1.14 iqtcl_GetCensusTable 63
5.5.1.15 iqtcl_GetProgramTableTagged 64
5.5.1.16 iqtcl_GetPidTableTagged 65
5.6 Connection Methods 66
5.6.1 Detailed Description 66
5.6.1.1 iqtcl_ConnectTo 67
5.6.1.2 iqtcl_OpenConnection 68
5.6.1.3 iqtcl_OpenConnectionWithMsgSize 69
5.6.1.4 iqtcl_CloseConnection 70
5.6.1.5 iqtcl_IsConnected 71
5.6.1.6 iqtcl_CloseSession 72
5.6.1.7 iqtcl_ResumeSession 73
5.6.1.8 iqtcl_CloseOldConnections 74
5.6.1.9 iqtcl_SaveSessionData 75
5.6.1.10 iqtcl_RetrieveSessionData 76
5.7 General Methods 77
5.7.1 Detailed Description 77
5.7.1.1 iqtcl_SetDefaultTimeout 78
5.7.1.2 iqtcl_GetLastError 79
5.7.1.3 iqtcl_GetAPIVersionString 80
5.7.1.4 iqtcl_GetMACAddress 81
5.7.1.5 iqtcl_GetCoreMode 82
5.7.1.6 iqtcl_SetCoreMode 83
5.7.1.7 iqtcl_GetTargetType 84
5.7.1.8 iqtcl_GetFirmwareMode 85
5.7.1.9 iqtcl_GetTargetInfo 86
5.7.1.10 iqtcl_GetTotalAvailableMemory 87
5.7.1.11 iqtcl_SendSyslogMsg 88
5.7.1.12 iqtcl_SendTargetSyslogMsg 89
5.7.1.13 iqtcl_SendCustomCommand 90
5.8 ARP Proxy Methods 91
5.8.1 Detailed Description 91
5.8.1.1 iqtcl_AddArpProxy 92

5.8.1.2 iqtcl_RemoveArpProxy 93
5.8.1.3 iqtcl_ClearArpProxyTable 94
5.8.1.4 iqtcl_AddArpProxyEx 95
5.8.1.5 iqtcl_RemoveArpProxyEx 96
5.9 IGMP Methods 97
5.9.1 Detailed Description 97
5.9.1.1 iqtcl_GetIGMPStatus 98
5.9.1.2 iqtcl_GetIGMPFirst 99
5.9.1.3 iqtcl_GetIGMPNext 100
5.9.1.4 iqtcl_IGMPJoin 101
5.9.1.5 iqtcl_IGMPLeave 102
5.9.1.6 iqtcl_StartIGMPLoop 103
5.9.1.7 iqtcl_StopIGMPLoop 104
5.9.1.8 iqtcl_ClearIGMP 105
5.10 STB Methods 106
5.10.1 Detailed Description 106
5.10.1.1 iqtcl_OpenSTBSession 107
5.10.1.2 iqtcl_CloseSTBSession 108
5.10.1.3 iqtcl_StartSTBSession 109
5.10.1.4 iqtcl_StopSTBSession 110
5.10.1.5 iqtcl_ClearSTBDefines 111
5.10.1.6 iqtcl_DefineNewSTB 112
5.10.1.7 iqtcl_GetSTBDataTable 113
5.11 License Methods 114
5.11.1 Detailed Description 114
5.11.1.1 iqtcl_IsLicenseValid 115
5.11.1.2 iqtcl_GetLicenseCount 116
5.12 MDI Methods 117
5.12.1 Detailed Description 117
5.12.1.1 iqtcl_StartMDIScan 118
5.12.1.2 iqtcl_StopMDIScan 119
5.12.1.3 iqtcl_GetMDIStatus 120
5.12.1.4 iqtcl_MDILockStream 121
5.12.1.5 iqtcl_MDIUnlockStream 122
5.13 QAM Methods 123
5.13.1 Detailed Description 123

5.13.1.1 iqtcl_QAMGetState 124
5.13.1.2 iqtcl_QAMGetActiveChannel 125
5.13.1.3 iqtcl_QAMGetStreamID 126
5.13.1.4 iqtcl_QAMGenerateAliasesFromLearn 127
5.13.1.5 iqtcl_QAMStartStream 128
5.13.1.6 iqtcl_QAMStartScan 129
5.13.1.7 iqtcl_IsQAMScanning 130
5.13.1.8 iqtcl_QAMStartLearn 131
5.13.1.9 iqtcl_QAMStartOp 132
5.13.1.10 iqtcl_QAMTuneAndStreamBySTBChanName 133
5.13.1.11 iqtcl_QAMTuneAndStreamBySTBChanNumber 134
5.13.1.12 iqtcl_QAMStopStream 135
5.13.1.13 iqtcl_QAMStopScan 136
5.13.1.14 iqtcl_QAMStopLearn 137
5.13.1.15 iqtcl_QAMStopOp 138
5.14 Record Methods 139
5.14.1 Detailed Description 139
5.14.1.1 iqtcl_OpenRecord 140
5.14.1.2 iqtcl_OpenStreamRecord 141
5.14.1.3 iqtcl_CloseRecord 142
5.14.1.4 iqtcl_StartRecord 143
5.14.1.5 iqtcl_StopRecord 144
5.14.1.6 iqtcl_UploadRecord 145
5.14.1.7 iqtcl_SaveUploadRecordToFile 146
5.14.1.8 iqtcl_GetRecordStatus 147
5.14.1.9 iqtcl_SetTriggerPos 148
5.14.1.10 iqtcl_AddTriggerCondition 149
5.14.1.11 iqtcl_RemoveTriggerCondition 150
5.14.1.12 iqtcl_GetTriggerStatus 151
5.14.1.13 iqtcl_IsRecordTriggered 152
5.14.1.14 iqtcl_StartRecord_TriggerPos 153
5.15 RVL Methods 154
5.15.1 Detailed Description 154
5.15.1.1 iqtcl_StartRVLScan 155
5.15.1.2 iqtcl_StopRVLScan 156
5.15.1.3 iqtcl_GetRVLStatus 157

5.15.1.4 iqtcl_RVLLockStream 158
5.15.1.5 iqtcl_RVLUnlockStream 159
5.16 Port Methods 160
5.16.1 Detailed Description 160
5.16.1.1 iqtcl_ClearPortCounters 161
5.16.1.2 iqtcl_GetPortStatus 162
5.16.1.3 iqtcl_GetPortCounters 163
5.16.1.4 iqtcl_GetPortCounterByIndex 164
5.16.1.5 iqtcl_GetPortCounterTableItem 165
5.16.1.6 iqtcl_GetPortCounterTable 166
5.17 Stimulus Methods 167
5.17.1 Detailed Description 167
5.17.1.1 iqtcl_GetStimulusStatus 168
5.17.1.2 iqtcl_OpenStimulus 169
5.17.1.3 iqtcl_OpenSmallStimulus 170
5.17.1.4 iqtcl_CloseStimulus 171
5.17.1.5 iqtcl_ReplicateStream 172
5.17.1.6 iqtcl_DownloadStimulusFile 173
5.17.1.7 iqtcl_LoadDefaultFile 174
5.17.1.8 iqtcl_SetBackgroundTraffic 175
5.17.1.9 iqtcl_DownloadLibpcapFile 176
5.17.1.10 iqtcl_StartStimulus 177
5.17.1.11 iqtcl_StopStimulus 178
5.17.1.12 iqtcl_GetStimDiscovery 179
5.17.1.13 iqtcl_StopStimDiscovery 180
5.17.1.14 iqtcl_SetTracer 181
5.17.1.15 iqtcl_ClearTracer 182
5.18 Stimulus Modification Methods 183
5.18.1 Detailed Description 183
5.18.1.1 iqtcl_SetXCount 184
5.18.1.2 iqtcl_SetIPDrops 185
5.18.1.3 iqtcl_SetIPJitter 186
5.18.1.4 iqtcl_SetDFJitter 187
5.18.1.5 iqtcl_SetPCRBitrate 188
5.18.1.6 iqtcl_SetBitrate 189
5.18.1.7 iqtcl_DropPid 190

5.18.1.8 iqtcl_StopIPDrops 191
5.18.1.9 iqtcl_StopIPJitter 192
5.18.1.10 iqtcl_StopDFJitter 193
5.18.1.11 iqtcl_StopDropPid 194

6 Data Structure Documentation 195

6.1 tALARMHANDLE Struct Reference 196
6.1.1 Field Documentation 196
6.1.1.1 tag 196
6.1.1.2 timestamp 196
6.1.1.3 handle 196
6.1.1.4 description 196
6.2 tALARMINFO Struct Reference 197
6.2.1 Field Documentation 197
6.2.1.1 tag 197
6.2.1.2 Id 197
6.2.1.3 alarmId 197
6.2.1.4 status 197
6.2.1.5 severity 197
6.2.1.6 streamId 197
6.2.1.7 threshold 197
6.2.1.8 value 197
6.2.1.9 timestamp 197
6.3 tALIASCONFIG Struct Reference 198
6.3.1 Field Documentation 198
6.3.1.1 tag 198
6.3.1.2 id 198
6.3.1.3 srcIpAddress 198
6.3.1.4 destIpAddress 199
6.3.1.5 srcIpMask 199
6.3.1.6 destIpMask 199
6.3.1.7 srcPort 199
6.3.1.8 destPort 199
6.3.1.9 name 199
6.3.1.10 igmpStatus 199
6.3.1.11 modType 199
6.3.1.12 mac 199

6.3.1.13 fmTemplate 199
6.3.1.14 fieldMask 199
6.3.1.15 bJoined 200
6.3.1.16 configStatus 200
6.3.1.17 ssrc 200
6.3.1.18 aliasType 200
6.3.1.19 charTemplate 200
6.3.1.20 vlanTci 200
6.3.1.21 videoType 200
6.3.1.22 tunerSdvType 200
6.3.1.23 tunerSdvMaxBw 200
6.3.1.24 tunerSdvDesc 200
6.3.1.25 intendedBitrate 200
6.3.1.26 intendedType 201
6.3.1.27 tsId 201
6.3.1.28 igmpSets 201
6.3.1.29 ports 201
6.4 tALIASNAME Struct Reference 202
6.4.1 Field Documentation 202
6.4.1.1 tag 202
6.4.1.2 alias 202
6.5 tASISTATUS Struct Reference 203
6.5.1 Field Documentation 203
6.5.1.1 tag 203
6.5.1.2 status 203
6.5.1.3 id 203
6.5.1.4 hostIP 203
6.6 tCENTRY Struct Reference 204
6.6.1 Field Documentation 204
6.6.1.1 tag 204
6.6.1.2 ID 204
6.6.1.3 timestamp 204
6.6.1.4 flags 204
6.6.1.5 streamType 204
6.6.1.6 hdrSize 204
6.6.1.7 payloadSize 204

6.6.1.8 bitrate 204
6.6.1.9 detectedBitrate 204
6.6.1.10 extFlags 204
6.7 tETHINFO Struct Reference 205
6.7.1 Field Documentation 205
6.7.1.1 tag 205
6.7.1.2 destMac 205
6.7.1.3 srcMac 205
6.8 tIGMPEVENT Struct Reference 206
6.8.1 Field Documentation 206
6.8.1.1 tag 206
6.8.1.2 handle 206
6.8.1.3 address 206
6.8.1.4 vlan 206
6.8.1.5 srcFilter 206
6.8.1.6 minTime 206
6.8.1.7 maxTime 206
6.8.1.8 lastTime 206
6.8.1.9 avgTime 206
6.8.1.10 result 206
6.8.1.11 join 206
6.8.1.12 leave 206
6.8.1.13 state 206
6.9 tIGMPGROUPS Struct Reference 207
6.9.1 Field Documentation 207
6.9.1.1 tag 207
6.9.1.2 address 207
6.10 tIGMPSTATS Struct Reference 208
6.10.1 Field Documentation 208
6.10.1.1 tag 208
6.10.1.2 handle 208
6.10.1.3 flags 208
6.10.1.4 address 208
6.10.1.5 minTime 208
6.10.1.6 maxTime 208
6.10.1.7 lastTime 208

6.10.1.8 avgTime 208
6.10.1.9 nSuccess 208
6.10.1.10 nFail 208
6.10.1.11 vlan 208
6.10.1.12 srcFilter 208
6.10.1.13 lastLveTime 208
6.11 tIGMPSTATUS Struct Reference 209
6.11.1 Field Documentation 209
6.11.1.1 tag 209
6.11.1.2 taskStatus 209
6.12 tIPINFO Struct Reference 210
6.12.1 Field Documentation 210
6.12.1.1 tag 210
6.12.1.2 srcIP 210
6.12.1.3 dstIP 210
6.12.1.4 srcPort 210
6.12.1.5 dstPort 210
6.12.1.6 protocol 210
6.12.1.7 tos 210
6.12.1.8 vlanID 210
6.13 tIXF18103CTRS Struct Reference 211
6.13.1 Field Documentation 211
6.13.1.1 tag 211
6.13.1.2 TxTotalOctets 211
6.13.1.3 TxMulticastFrames 211
6.13.1.4 TxBroadcastFrames 211
6.13.1.5 TxTotalFrames 212
6.13.1.6 TxSizeFrames 212
6.13.1.7 TxVLANFrames 212
6.13.1.8 TxPAUSECtrlFrames 212
6.13.1.9 TxUnicastFrames 212
6.13.1.10 TxMACCtrlFrames 212
6.13.1.11 RxTotalOctets 212
6.13.1.12 RxMulticastFrames 212
6.13.1.13 RxBroadcastFrames 212
6.13.1.14 RxTotalFrames 212

6.13.1.15 RxSizeFrames 212
6.13.1.16 RxVLANFrames 213
6.13.1.17 RxPAUSECtrlFrames 213
6.13.1.18 RxUnicastFrames 213
6.13.1.19 RxMACCtrlFrames 213
6.13.1.20 RxEthUndersized 213
6.13.1.21 RxEthOversized 213
6.13.1.22 RxEthOctets 213
6.13.1.23 RxEthPkts 213
6.13.1.24 RxEthFragments 213
6.13.1.25 RxEthJabbers 213
6.13.1.26 RxEthFcs 213
6.13.1.27 TxTotalBytes 214
6.13.1.28 RxTotalBytes 214
6.13.1.29 RxBadFrames 214
6.13.1.30 RxGoodFrames 214
6.14 tMDIINFO Struct Reference 215
6.14.1 Field Documentation 215
6.14.1.1 tag 215
6.14.1.2 nSamples 215
6.14.1.3 dfMin 215
6.14.1.4 dfMax 215
6.14.1.5 dfCurrent 215
6.14.1.6 dfAvg 215
6.14.1.7 dfTotal 215
6.14.1.8 mlMin 215
6.14.1.9 mlMax 215
6.14.1.10 mlCurrent 215
6.14.1.11 mlAvg 215
6.14.1.12 mlTotal 215
6.14.1.13 ml15 215
6.14.1.14 ml24 215
6.14.1.15 vbMin 215
6.14.1.16 vbMax 215
6.14.1.17 vbCurrent 215
6.14.1.18 vbAvg 215

6.14.1.19 vbTotal 215
6.15 tMPEG2INFO Struct Reference 216
6.15.1 Field Documentation 216
6.15.1.1 tag 216
6.15.1.2 networkPid 216
6.15.1.3 nPrograms 216
6.15.1.4 patVersion 216
6.15.1.5 mtspSize 216
6.15.1.6 nProgramAdded 216
6.15.1.7 nProgramRemoved 216
6.15.1.8 tsId 216
6.16 tMPEG2PID Struct Reference 217
6.16.1 Field Documentation 217
6.16.1.1 tag 217
6.16.1.2 pid 217
6.16.1.3 type 217
6.16.1.4 flags 217
6.16.1.5 nSamples 217
6.16.1.6 pbrMin 217
6.16.1.7 pbrMax 217
6.16.1.8 pbrCurrent 217
6.16.1.9 pbrAvg 217
6.16.1.10 pbrTotal 217
6.16.1.11 ccErrCurrent 217
6.16.1.12 ccErrTotal 217
6.16.1.13 extFlags 217
6.16.1.14 outagePd 217
6.16.1.15 lossRatio 217
6.16.1.16 stateTime 217
6.16.1.17 outages 217
6.16.1.18 language 217
6.16.1.19 misc 217
6.16.1.20 duplicate 217
6.17 tMPEG2PROGRAM Struct Reference 218
6.17.1 Field Documentation 218
6.17.1.1 tag 218

6.17.1.2 nChannel 218
6.17.1.3 nPids 218
6.17.1.4 name 218
6.17.1.5 aliasName 218
6.17.1.6 chanNumber 218
6.17.1.7 progStatus 218
6.17.1.8 alarmPids 218
6.17.1.9 deviceRef 218
6.17.1.10 flags 218
6.17.1.11 curBitrate 218
6.17.1.12 stateTime 218
6.17.1.13 providerName 218
6.17.1.14 totLoss 218
6.17.1.15 curMlr 218
6.17.1.16 firstPidIndex 218
6.17.1.17 crc 218
6.18 tMTSPSTATS Struct Reference 219
6.18.1 Field Documentation 219
6.18.1.1 tag 219
6.18.1.2 ccErrTotal 219
6.18.1.3 ccErrCurrent 219
6.18.1.4 syncError 219
6.18.1.5 syncErrorTotal 219
6.18.1.6 totalPids 219
6.18.1.7 monPids 219
6.18.1.8 almPids 219
6.18.1.9 future 219
6.18.1.10 nSamples 219
6.18.1.11 pbrMin 219
6.18.1.12 pbrMax 219
6.18.1.13 pbrCurrent 219
6.18.1.14 pbrAvg 219
6.18.1.15 pbrTotal 219
6.19 tNAMETAG Struct Reference 220
6.19.1 Field Documentation 220
6.19.1.1 tag 220

6.19.1.2 name 220
6.20 tNemoCtrs Struct Reference 221
6.20.1 Field Documentation 221
6.20.1.1 RxOctetsTotal 221
6.20.1.2 RxPacketsTotal 221
6.20.1.3 RxPacketsUnicast 221
6.20.1.4 RxPacketsMulticast 221
6.20.1.5 RxPacketsBroadcast 221
6.20.1.6 RxPackets 221
6.20.1.7 TxOctetsTotal 221
6.20.1.8 TxPacketsTotal 222
6.20.1.9 TxPacketsUnicast 222
6.20.1.10 TxPacketsMulticast 222
6.20.1.11 TxPacketsBroadcast 222
6.20.1.12 TxPackets 222
6.20.1.13 RxCrcError 222
6.20.1.14 RxErrors 222
6.21 tNEMOCTRS Struct Reference 223
6.21.1 Field Documentation 223
6.21.1.1 tag 223
6.21.1.2 ctrs 223
6.22 tOldMPEG2PROGRAM Struct Reference 224
6.22.1 Field Documentation 224
6.22.1.1 tag 224
6.22.1.2 nChannel 224
6.22.1.3 pmtPid 224
6.22.1.4 pcrPid 224
6.22.1.5 nPids 224
6.23 tPMPIDSTATS Struct Reference 225
6.23.1 Field Documentation 225
6.23.1.1 tag 225
6.23.1.2 maxBr 225
6.23.1.3 minBr 225
6.23.1.4 loss 225
6.23.1.5 alarms 225
6.23.1.6 ess 225

6.23.1.7 ivlFaults 225
6.23.1.8 avgBr 225
6.23.1.9 outages 225
6.23.1.10 outagePd 225
6.23.1.11 ivlMask 225
6.23.1.12 ivlHist 225
6.23.1.13 ivlState 225
6.24 tPMPIDTOTALSTATS Struct Reference 226
6.24.1 Field Documentation 226
6.24.1.1 tag 226
6.24.1.2 stopTime 226
6.24.1.3 totalBr 226
6.24.1.4 mlt24 226
6.24.1.5 stateFlags 226
6.24.1.6 totAlarms 226
6.24.1.7 totEss 226
6.24.1.8 totOutagePd 226
6.25 tPMPROGRAMIVLSTATS Struct Reference 227
6.25.1 Field Documentation 227
6.25.1.1 tag 227
6.25.1.2 maxBr 227
6.25.1.3 minBr 227
6.25.1.4 ess 228
6.25.1.5 outPids 228
6.25.1.6 alarms 228
6.25.1.7 maxMlr 228
6.25.1.8 mls15 228
6.25.1.9 mlt15 228
6.25.1.10 ivlMask 228
6.25.1.11 ivlFaults 228
6.25.1.12 ivlHist 228
6.25.1.13 progStatus 228
6.25.1.14 almPids 228
6.25.1.15 monPids 229
6.25.1.16 monOutPids 229
6.25.1.17 outages 229

6.25.1.18 outagePd 229
6.25.1.19 totOutagePd 229
6.25.1.20 totEss 229
6.25.1.21 totScteEvts 229
6.25.1.22 monitors 229
6.25.1.23 ivlFlags 229
6.25.1.24 avgBr 229
6.25.1.25 lastProgStatus 229
6.25.1.26 lossAlarms 230
6.25.1.27 totMls 230
6.25.1.28 maxMLp 230
6.25.1.29 minMLd 230
6.25.1.30 stateChanges 230
6.26 tPMPROGRAMTOTSTATS Struct Reference 231
6.26.1 Field Documentation 231
6.26.1.1 tag 231
6.26.1.2 mlt24 231
6.26.1.3 ess 231
6.26.1.4 mls24 231
6.26.1.5 scteEvtTime 231
6.26.1.6 stateCount 231
6.26.1.7 pidStateCount 231
6.26.1.8 totScteEvts 231
6.26.1.9 maxMLp 231
6.26.1.10 curMLp 231
6.26.1.11 curMLd 231
6.26.1.12 totEss 231
6.26.1.13 totMLT 231
6.26.1.14 totMLS 231
6.26.1.15 totOutagePd 231
6.26.1.16 totMaxMLp 231
6.26.1.17 totMinMLd 231
6.27 tPMSTREAMGRAPHMETRICS Struct Reference 232
6.27.1 Field Documentation 233
6.27.1.1 tag 233
6.27.1.2 streamId 233

6.27.1.3 ivlTime 233
6.27.1.4 minBitRate 233
6.27.1.5 maxBitRate 233
6.27.1.6 pktLoss 233
6.27.1.7 mdiDf 233
6.27.1.8 rtpLd 233
6.27.1.9 Mls 233
6.27.1.10 Ess 233
6.27.1.11 Sess 233
6.27.1.12 Pess 234
6.27.1.13 starts 234
6.27.1.14 lastFlowPayldStatus 234
6.27.1.15 minPktRate 234
6.27.1.16 maxPktRate 234
6.27.1.17 minVBuffer 234
6.27.1.18 faultStatus 234
6.27.1.19 outagePd 234
6.27.1.20 faultTime 234
6.27.1.21 maxVBuffer 234
6.27.1.22 minVTsb 234
6.27.1.23 maxVTsb 235
6.27.1.24 lossProgCount 235
6.27.1.25 monPrograms 235
6.27.1.26 fltPrograms 235
6.27.1.27 monTsPids 235
6.27.1.28 fltTsPids 235
6.27.1.29 extFlags 235
6.27.1.30 flags 235
6.27.1.31 eMask 235
6.27.1.32 eFaults 235
6.27.1.33 eHistory 235
6.27.1.34 rtpLoss 236
6.27.1.35 rtpLossEvts 236
6.27.1.36 rtplP 236
6.27.1.37 retryReqs 236
6.27.1.38 retryFills 236

6.27.1.39 usrFeebacks 236
6.27.1.40 rtpLs 236
6.27.1.41 flowPayldStatus 236
6.27.1.42 lossPercent 236
6.27.1.43 eStateChanges 236
6.28 tPMSTREAMMETRICS Struct Reference 237
6.28.1 Field Documentation 237
6.28.1.1 tag 237
6.28.1.2 streamId 237
6.28.1.3 Sdps 237
6.28.1.4 Mls 237
6.28.1.5 Ess 237
6.28.1.6 Sess 237
6.28.1.7 Uass 237
6.28.1.8 starts 237
6.28.1.9 faults 237
6.28.1.10 flags 237
6.29 tPMSTREAMTOTALMETRICS Struct Reference 238
6.29.1 Field Documentation 238
6.29.1.1 tag 238
6.29.1.2 streamId 238
6.29.1.3 progNoAliasCnt 238
6.29.1.4 progAliases 239
6.29.1.5 Mls24 239
6.29.1.6 Ess 239
6.29.1.7 Sess 239
6.29.1.8 Pess 239
6.29.1.9 Actss 239
6.29.1.10 Totss 239
6.29.1.11 Totsts 239
6.29.1.12 pktLoss 239
6.29.1.13 outagePd 239
6.29.1.14 outageCt 239
6.29.1.15 Mls 240
6.29.1.16 usrQos 240
6.29.1.17 ls24 240

6.29.1.18 rtpLoss24 240
6.29.1.19 lossPercent 240
6.29.1.20 stateTime 240
6.29.1.21 progStateCount 240
6.29.1.22 flowStateCount 240
6.29.1.23 ledToFaultMap 240
6.29.1.24 mgtId 240
6.29.1.25 totOutagePd 240
6.29.1.26 totMloss 241
6.29.1.27 totEss 241
6.29.1.28 totPess 241
6.30 tPMSYSTEMMETRICS Struct Reference 242
6.30.1 Field Documentation 242
6.30.1.1 tag 242
6.30.1.2 ivlTime 242
6.30.1.3 tNewStreams 242
6.30.1.4 tBadStreams 243
6.30.1.5 tMaxStreams 243
6.30.1.6 tMinStreams 243
6.30.1.7 mediaLoss 243
6.30.1.8 fltMapChanged 243
6.30.1.9 blueStreams 243
6.30.1.10 greyStreams 243
6.30.1.11 greenStreams 243
6.30.1.12 redStreams 243
6.30.1.13 orangeStreams 243
6.30.1.14 util 243
6.30.1.15 activeStreams 244
6.30.1.16 usrQos 244
6.30.1.17 retryReqs 244
6.30.1.18 evtsP0 244
6.30.1.19 evtsP1 244
6.30.1.20 retryFills 244
6.30.1.21 mls 244
6.30.1.22 maxLp 244
6.30.1.23 minLp 244

6.30.1.24 lpErrors 244
6.30.1.25 minLd 244
6.30.1.26 ipLoss 245
6.30.1.27 bcastStreams 245
6.30.1.28 evtsP2 245
6.30.1.29 systemStatus 245
6.30.1.30 flags 245
6.30.1.31 trapSentRate 245
6.30.1.32 timestamp 245
6.31 tRTPINFO Struct Reference 246
6.31.1 Field Documentation 246
6.31.1.1 tag 246
6.31.1.2 payloadType 246
6.32 tRTPSTATS Struct Reference 247
6.32.1 Field Documentation 247
6.32.1.1 tag 247
6.32.1.2 seqErrTotal 247
6.32.1.3 seqErrCurrent 247
6.32.1.4 ldMin 247
6.32.1.5 lpMax 247
6.32.1.6 ldCurrent 247
6.32.1.7 lpCurrent 247
6.32.1.8 ldErrors 247
6.32.1.9 lpErrors 247
6.32.1.10 lossDuration 247
6.32.1.11 lossEvtCurrent 247
6.32.1.12 lossEvtTotal 247
6.32.1.13 lossPercent 247
6.32.1.14 dupCurrent 247
6.32.1.15 dupTotal 247
6.32.1.16 oosCurrent 247
6.32.1.17 oosTotal 247
6.33 tSTREAMSTATS Struct Reference 248
6.33.1 Field Documentation 248
6.33.1.1 tag 248
6.33.1.2 nSamples 248

6.33.1.3 pktSizeMin . . . 248
6.33.1.4 pktSizeMax . . . 249
6.33.1.5 lbrMin . . . 249
6.33.1.6 lbrMax . . . 249
6.33.1.7 lbrCurrent . . . 249
6.33.1.8 lbrAvg . . . 249
6.33.1.9 lbrTotal . . . 249
6.33.1.10 mbrMin . . . 249
6.33.1.11 mbrMax . . . 249
6.33.1.12 mbrCurrent . . . 249
6.33.1.13 mbrAvg . . . 249
6.33.1.14 mbrTotal . . . 249
6.33.1.15 utilMin . . . 250
6.33.1.16 utilMax . . . 250
6.33.1.17 utilCurrent . . . 250
6.33.1.18 utilAvg . . . 250
6.33.1.19 utilTotal . . . 250
6.33.1.20 faultStatus . . . 250
6.33.1.21 faultMap . . . 250
6.33.1.22 faultHistory . . . 250
6.33.1.23 faultTime . . . 250
6.33.1.24 decayCount . . . 250
6.33.1.25 tos . . . 250
6.33.1.26 userFeedback . . . 251
6.33.1.27 pktMin . . . 251
6.33.1.28 pktMax . . . 251
6.33.1.29 pktCurrent . . . 251
6.33.1.30 pktAvg . . . 251
6.33.1.31 pktTotal . . . 251
6.34 tTAG Struct Reference . . . 252
6.34.1 Field Documentation . . . 252
6.34.1.1 type . . . 252
6.34.1.2 size . . . 252
6.35 tTAPSTATUS Struct Reference . . . 253
6.35.1 Field Documentation . . . 253
6.35.1.1 tag . . . 253

6.35.1.2 status 253
6.35.1.3 id 253
6.35.1.4 hostIP 253
6.36 tTARGETINFO Struct Reference 254
6.36.1 Field Documentation 254
6.36.1.1 tag 254
6.36.1.2 targetMode 254
6.36.1.3 timeDate 254
6.36.1.4 targetName 254
6.36.1.5 targetLocation 254
6.36.1.6 targetContact 254
6.37 tTRITENCTRS Struct Reference 255
6.37.1 Field Documentation 255
6.37.1.1 tag 255
6.37.1.2 ctrs 255
6.38 tTritenCtrs Struct Reference 256
6.38.1 Field Documentation 257
6.38.1.1 RxOctetsGood 257
6.38.1.2 RxOctetsBad 257
6.38.1.3 RxPacketsUnicast 257
6.38.1.4 RxPacketsMulticast 257
6.38.1.5 RxPacketsBroadcast 257
6.38.1.6 RxPackets 257
6.38.1.7 RxFcsErrors 257
6.38.1.8 RxTagged 257
6.38.1.9 RxDataErrors 257
6.38.1.10 RxAlignErrors 257
6.38.1.11 RxLongErrors 257
6.38.1.12 RxJabberErrors 258
6.38.1.13 RxPauseControl 258
6.38.1.14 RxUnknownControl 258
6.38.1.15 RxVeryLongErrors 258
6.38.1.16 RxRuntErrors 258
6.38.1.17 RxShortErrors 258
6.38.1.18 CarrierExtendErrors 258
6.38.1.19 RxSequenceErrors 258

6.38.1.20 RxSymbolErrors 258
6.38.1.21 RxTotalOctets 258
6.38.1.22 RxTotalPackets 258
6.38.1.23 RxTotalErrors 259
6.38.1.24 TxOctetsGood 259
6.38.1.25 TxOctetsBad 259
6.38.1.26 TxPacketsUnicast 259
6.38.1.27 TxPacketsMulticast 259
6.38.1.28 TxPacketsBroadcast 259
6.38.1.29 TxPackets 259
6.38.1.30 TxDeffered 259
6.38.1.31 TxTotalCollisions 259
6.38.1.32 TxSingleCollisions 259
6.38.1.33 TxMultipleCollisions 259
6.38.1.34 TxLateCollisions 260
6.38.1.35 TxExcessiveCollisionErrors 260
6.38.1.36 TxExcessiveDefferalErrors 260
6.38.1.37 TxExcessiveLengthDrop 260
6.38.1.38 TxUnderrun 260
6.38.1.39 TxTagged 260
6.38.1.40 TxFcsErrors 260
6.38.1.41 TxPauseFrames 260
6.38.1.42 TxFlowControlCollisions 260
6.38.1.43 TxTotalOctets 260
6.38.1.44 TxTotalPackets 260
6.38.1.45 TxTotalErrors 261
6.38.1.46 TapPacketsDropped 261
6.38.1.47 TapBytesDropped 261
6.39 tTUNERRFMAP Struct Reference 262
6.39.1 Field Documentation 262
6.39.1.1 tag 262
6.39.1.2 num 262
6.39.1.3 freq 262
6.39.1.4 symbolRate 262
6.40 tTUNERSTATS Struct Reference 263
6.40.1 Field Documentation 263

6.40.1.1 tag 263
6.40.1.2 freq 263
6.40.1.3 chan 264
6.40.1.4 signal 264
6.40.1.5 mod 264
6.40.1.6 id 264
6.40.1.7 active 264
6.40.1.8 nSamples 264
6.40.1.9 snrMin 264
6.40.1.10 snrMax 264
6.40.1.11 snrCurrent 264
6.40.1.12 snrAvg 264
6.40.1.13 snrTotal 264
6.40.1.14 rsUcCurrent 265
6.40.1.15 rsUcTotal 265
6.40.1.16 rsCoCurrent 265
6.40.1.17 rsCoTotal 265
6.40.1.18 chanPre 265
6.40.1.19 rsUcPercent 265
6.40.1.20 rsCoPercent 265
6.40.1.21 rsBytesTotal 265
6.40.1.22 tuner 265
6.40.1.23 align 265
6.40.1.24 flags 265
6.40.1.25 nSamplesRxPwr 266
6.40.1.26 rxPwrMin 266
6.40.1.27 rxPwrMax 266
6.40.1.28 rxPwrCurrent 266
6.40.1.29 rxPwrAvg 266
6.40.1.30 rxPwrTotal 266
6.40.1.31 nSamplesBer 266
6.40.1.32 berPreMin 266
6.40.1.33 berPreMax 266
6.40.1.34 berPreCurrent 266
6.40.1.35 berPreAvg 266
6.40.1.36 berPreTotal 267

6.40.1.37 berPostMin 267
6.40.1.38 berPostMax 267
6.40.1.39 berPostCurrent 267
6.40.1.40 berPostAvg 267
6.40.1.41 berPostTotal 267

Chapter 1

IneoQuest Tcl API Reference Manual

1.1 General API Information

The IneoQuest Tcl support for controlling Singulus boxes consists of two libraries.

- The C-API library provides support for controlling and communicating with IneoQuest Singulus boxes.
- The Tcl-API provides a bridge to allow Tcl scripts to make calls into the C-API library.

The Tcl-API provides different levels of abstraction built atop the “bare-metal” firmware commands supported by the C-API. The Tcl-API methods mostly correspond to C methods, but provide an easier programming methodology than structured C. There also exists a Command-Line Interface Framework.

In addition to these libraries, the Tcl-API methods are documented in the pdf file installed into the /doc directory. Several Tcl scripts are installed into the /scripts directory. These scripts provide a basic example on how to use the Tcl-API.

Chapter 2

Command Line Interface for IneoQuest Tcl API

The Command-Line Interface (CLI) provides a simple and automatable means to run standardized tests for IneoQuest devices.

The CLI wraps around the lower-level Tcl methods, aggregates them into common control flows, and further collects these tasks into more user-friendly, and market-specific tests for automated testing systems.

```
---------------------------------------------------------------------
| Host (Windows, Linux, Solaris server)                             |
---------------------------------------------------------------------
...                                                               ...
---------------------------------------------------------------------
| CLI Tests        |                                                |
|   Stimulus mode  |   DropPID                                      |
|                  |   GeneralStimTest                              |
|                  |                                                |
|   Analysis mode  |   CheckFLows                                   |
|                  |   TriggerCapture                               |
|                  |  ... etc ...                                   |
------------------|--------------------------------------------------
| CLI Methods      |  ConfigDutyCycles                              |
|                  |  CauseDropPID                                  |
|                  |  UndoDropPID                                   |
|                  |  CauseIPJitter                                 |
|                  |  ... etc ...                                   |
---------------------------------------------------------------------
| Generic procs    |    ConnectTo     | CloseOldConnections    | etc   |
---------------------------------------------------------------------
| Tcl API methods  | iqtcl_ConnectTo  | iqtcl_CloseConnection  | etc   |
---------------------------------------------------------------------
| C API methods    |   Connect        | Close                  | etc   |
---------------------------------------------------------------------
...                                                               ...
---------------------------------------------------------------------
| Target (G1-T in Stimulus or Analysis mode; IP/QAM Crickets, etc)  |
---------------------------------------------------------------------
```

The CLI supports both Stimulus and Analysis modes of operation to generate video flows and gather information about video flows, respectively, and provides Analysis Reports and Stimulus Tests.

The overall test setup used in the CLI is to have a stimulus engine providing video flows through a set of devices under test to an analysis engine. Both engines are controlled via a host running the Tcl-API CLI.

```
  -------------------      --------------      -------------------
  | STIMULUS ENGINE | ---> |  DEVICE(S) | ---> | ANALYSIS ENGINE |
  |                 |      |  UNDER     |      |                 |
  | G1-T            |      |  TEST      |      | IP Cricket      |
  | G10             |      --------------      | QAM Cricket     |
  | MediaStimulus   |                          | MediaAnalyzer   |
  | etc             |                          | etc             |
  -------------------                          -------------------
```

The Stimulus Tests are a collection of standard examples typically needed by IneoQuest customers to perform tests on and measurements of their device(s) under test (DUT). The fundamental context of a test is to perform some action in time. The action itself is the event, the timing is split into the time in in which the event is active (called the event duration) and the time in which it is inactive. Repeating a test, the total active/inactive event time required is periodic (called the event period). The total test time can be a single or multiple time periods.

In an engineering context, this timing is described by duty cycles and can be described by the figure below:

```
               |      |                      |                      |       |
               |<---->|<- EVENT PERIOD (Y) ->|<- EVENT PERIOD (Y) ->|       |
               |      |                      |                      |       |
               | Wait |                      |                      |       |
               | Time |                      |                      |       |
Active State   |      |__________            |__________            |       |
               |      |<-------->|           |<-------->|           |       |
               |      |   Event  |           |   Event  |           |       |
Inactive State |_...._| Duration |___________| Duration |___________|__...__|
               |      |    (X)               |    (X)               |       |
               |<-------------------- TOTAL TEST TIME --------------------->|
```

Variability in Stimulus Tests is provided by configuration files which set the parameters and act as a way to build, deploy, and repeatedly run standard tests on multiple systems.

For Analysis mode reporting, the parameters include which fields to query during the analysis. IneoQuest hardware utilizes the concept of a Flow Census, which aggregates the metrics associated with video flows being analyzed. The Tcl-API will query the hardware for data from the Census and provide the processed information via analysis procedures.

2.1 Stimulus-Mode Configuration Files

This section discusses the format for a Stimulus Configuration File. When the CLI framework is in Stimulus-mode, configuration files specify details of the stimuli. These details include the:

- Target(s) running the Stimulus engine
- Duty Cycle parameters (because stimulus sessions are connection-based and not threaded, duty cycles are per-target)
- Flow encapsulation parameters
- PID information (for Stim Tests performing actions on specific PIDs)

Stimulus targets are given as IP addresses. Each target is associated with a duty cycle, the format of which is: `duration=X period=Y` where `X` is the active event duration and `Y` is the event period of the cycle.

The format of a flow defines space-delimited encapsulation parameters as follows:

- srcMac: Source MAC address
- dstMac: Destination MAC address
- vlanID: VLAN ID
- TosField: TOS Field
- srcIP: Source IP address
- dstIP: Destination IP address
- srcPort: Source port
- dstPort: Destination port

- RtpType: RTP type
- EncType: Encapsulation type
  - 0: Eth2/IP/UDP encapsulation
  - 1: Eth2/IP/UDP/RTP encapsulation
  - 2: Eth2-VLAN/IP/UDP encapsulation
  - 3: Eth2-VLAN/IP/UDP/RTP encapsulation
- StimFile: Stimulus File (TS)
- StimPort: Stimulus Port (0 or 1)
- numStreams: Number of streams to replicate for stim

Example:

The following would create a stimulus of 24 flows across Port 0 of the transport stream file SPTS.ts from a source MAC of 00:08:D4:00:00:01, source IP address of 10.0.17.76, source port 8888 to a destination MAC of 00:08:D4:00:00:02, destination IP address of 10.0.17.87, destination port of 8890:

```
0008D4000001 0008D4000002 0 0 10.0.17.76 10.0.17.87 8888 8890 0 0 SPTS.ts 0 24
```

Comments are lines beginning with #.

The following is an example Stimulus-mode configuration file:

```
# Comment Line
TARGETS: 192.168.1.2
    DutyCycle: duration=15 period=60
    FLOW: 000000000001 000000000002 0 0 0.0.0.1 0.0.0.2 1024 2048 0 0 SPTS.ts 0 9
            PID: 4130 4131
            DropIP: 10 in 100
            DFJitter: start=400 stop=1200 stride=400
            IPJitter: 10 in 1000

    FLOW: 000000000003 000000000004 0 0 0.0.0.3 0.0.0.4 1024 2048 0 0 SPTS.ts 1 7
            DropIP: 10 in 100
            PID: 4131
            DFJitter: start=400 stop=1000 stride=50
            IPJitter: 10 in 100
```

2.2 Analysis-Mode Configuration Files

This section discusses the format of an Analysis Configuration File. When the CLI framework is in Analysis-mode, configuration files specify details of the analyses. These details depend on whether the Analysis target is monitoring video feeds delivered by RF or IP. The type of analysis needs to be described on the first line of the analysis configuration file.

A feature of this analysis framework is to confirm quality of video delivery within certain tolerance limits. Similar to engineering data-sheets, these limits can be described in terms of a minimum value, maximum value, and expected value with a threshold plus or minus the expected value. Therefore, the analysis-mode configuration file uses a key-value pairing and individual fields specifying quality. For PID-level parameters, the key-value described the PID number, and the fields describe data available from the probe's census. Fields below the min, above the max, or outside of the expected plus-or-minus the threshold percentage effect Fail conditions - otherwise, the test is a Pass.

Currently the PID-level metrics supported are the `BitRate, OutageCount,` and `OutageSec` fields.

If the analysis is of RF type, you can group channels and specify how much time to spend scanning each channel in the group. If IP type, the Flow parameters are specified.

An example FP-type configuration file running on 5 QAM probes, scanning different channels for different scan times, with associated PID-level datasheet parameters follows (indentation is important, indents are tabs):

```
type=rf

TARGETS: 192.168.2.1 192.168.2.2 192.168.2.3
        DefaultCaptureTime: 10

        CHANNELS: 45 48 72 73 74
                ScanTime: 30
                CaptureTime: 60

        PID: 1984
                BitRate: min=754000 exp=1925112 th=.05 max=5712192
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
        PID: 1985
                BitRate: min=19400 exp=195520 th=.05 max=200000
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0

        CHANNELS: 92 93 94 100 101 102 103 104
                ScanTime: 45
                CaptureTime: 60

        PID: 2048
                BitRate: min=754000 exp=1925112 th=.05 max=5712192
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
        PID: 2049
                BitRate: min=19400 exp=195520 th=.05 max=200000
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0

TARGETS: 192.168.2.4 192.168.2.5
        DefaultCaptureTime: 10

        CHANNELS: 76 77 78 79 82 83 84 85 86 87 88 89
                ScanTime: 60
                CaptureTime: 60

        PID: 1984
                BitRate: min=754000 exp=1925112 th=.05 max=5712192
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
        PID: 1985
                BitRate: min=19400 exp=195520 th=.05 max=200000
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
        CHANNELS: 109 110 111 112 113 114 115 116 117 118 119 120 121 122
                ScanTime: 30
                CaptureTime: 60

        PID: 2010
                BitRate: min=754000 exp=1925112 th=.05 max=5712192
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
        PID: 2011
                BitRate: min=19400 exp=195520 th=.05 max=200000
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
```

For IP-type analysis, the format of the Flow parameter is similar to that of a Stimulus configuration file, but only requires the IP parameters (source IP, destination IP, sort port, destination port).

For example:

```
type=ip

TARGETS: 192.168.3.1 192.168.3.2
        DefaultCaptureTime: 10

        FLOW: 0.0.0.1 0.0.0.2 1024 2048
                ScanTime: 5
                CaptureTime: 60

        PID: 4130
                BitRate: min=754000 exp=1925112 th=.05 max=5712192
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
        PID: 4131
                BitRate: min=19400 exp=195520 th=.05 max=200000
                OutageCount: min=0 exp=0 th=0.0 max=0
                OutageSec: min=0 exp=0 th=0.0 max=0
```

2.3 CLI Simulus-Mode Examples

This section presents some examples for the CLI framework when controlling Stimulus Tests. The CLI is intended to run on a command line session and to take command-line arguments specifying the mode to run in, the test or report to run, and any other options. For Stimlus-mode, commands should be of the form:

```
./CLI.tcl stimulus [TEST] [CONFIG FILE] [TOTAL TEST TIME (s)] [WAIT TIME (s)]
```

The total test and wait times are described above in the figure showing duty cycles.

You can see which tests are supported by running this command:

```
./CLI.tcl stimulus list
```

Currently, the following tests are supported:

- DropPID - drops a specific program ID from the flow

- PassPID - passes a PID
- DropIP - drops N IP packets in M total packets
- IPJitter - introduces IP jitter
- DFJitter - introduces DF jitter
- GeneralTest - runs through above tests, escalating monitor states

2.4 CLI Analysis-Mode Examples

This section presents some examples for the CLI framework when controlling Analysis Reports. The CLI is intended to run on a command line session and to take command-line arguments specifying the mode to run in, the test or report to run, and any other options. For Stimlus-mode, commands should be of the form:

```
./CLI.tcl analysis [TEST] [CONFIG FILE]
```

You can see which tests are supported by running this command:

```
./CLI.tcl analysis list
```

Currently, the following tests are supported:

- CheckFlows - check flow parameters in a data-sheet like form
- TriggerCaptureActive - capture first flow or active channel

2.4.1 Check Flow Parameters

The IQ probe hardware supports monitoring video feeds that are delivered via RF or IP, described above in Analysis-Mode Configuration Files. Here is example output of for an IP Cricket. In this example, PIDs 4130 and 4131 are dropped out of the flow.

```
[mgb cli]$ ./CLI.tcl analysis CheckFlows ./config/IPExample.txt
Running CheckFlows on 1 probe(s)
 Program 1/1: Pid 4130 CurrBR value 0: less than min value of 754000
 Program 1/1: Pid 4130 OutageSec value 13: more than max value of 0
 Program 1/1: Pid 4131 CurrBR value 0: less than min value of 19400
192.168.3.1 Failed CheckFlows
CheckFlows complete: 0/1 passed
```

2.4.2 Trigger Video Captures

For an RF feed, the Trigger Capture will immediately start capturing video on the active channel (in this case channel 80, 81):

```
[mgb cli]$ ./CLI.tcl analysis TriggerCaptureActive ./config/QAMExample.txt
Running TriggerCaptureActive on 2 probe(s)
```

```
->CAPTURE TRIGGERED (10000 ms) on 192.168.2.1
  Time   500 ms:  Post-buffer is 0.000% complete
  Time  1500 ms:  Post-buffer is 0.680% complete
  Time  2500 ms:  Post-buffer is 1.680% complete
  Time  3500 ms:  Post-buffer is 2.380% complete
  Time  4500 ms:  Post-buffer is 3.101% complete
  Time  5500 ms:  Post-buffer is 3.938% complete
  Time  6500 ms:  Post-buffer is 4.747% complete
  Time  7500 ms:  Post-buffer is 5.556% complete
  Time  8500 ms:  Post-buffer is 6.365% complete
  Time  9500 ms:  Post-buffer is 7.175% complete
done
Stopping record.........done

Uploading recording.....done
192.168.2.1 Passed TriggerCaptureActive

->CAPTURE TRIGGERED (10000 ms) on 192.168.2.2
  Time   600 ms:  Post-buffer is 0.057% complete
  Time  1600 ms:  Post-buffer is 0.709% complete
  Time  2600 ms:  Post-buffer is 1.360% complete
  Time  3600 ms:  Post-buffer is 2.012% complete
  Time  4600 ms:  Post-buffer is 2.663% complete
  Time  5600 ms:  Post-buffer is 3.315% complete
  Time  6600 ms:  Post-buffer is 3.966% complete
  Time  7600 ms:  Post-buffer is 4.615% complete
  Time  8600 ms:  Post-buffer is 5.267% complete
  Time  9600 ms:  Post-buffer is 5.918% complete
done
Stopping record.........done

Uploading recording.....done
192.168.2.2 Passed TriggerCaptureActive
TriggerCaptureActive complete: 2/2 passed

[mgb cli]$ ls *.ts
TriggerRecording-192.168.2.1-Chan_80.ts
TriggerRecording-192.168.2.2-Chan_81.ts
```

Chapter 3

Tcl Documentation Revision History

3.1 Revision History

2005-2006	Date	Description
1.0	January 22, 2005	Initial Release
2.0	May 31, 2005	Tcl API Version 2.0 specification
2.1	July 28, 2005	Tcl API Version 2.1 specification
2.2	August 8, 2005	Documented 'Tracer' to 'Default' name change
2.3	August 11, 2005	Added STB methods and documentation
2.4	August 31, 2005	Added iqtcl_ClearSTBDefines method
2.5	September 8, 2005	Added iqtcl_ClearAlarms method
2.6	September 21, 2005	Added iqtcl_SendSyslogMsg method
2.7	October 3, 2005	Added iqtcl_AddArpProxy, iqtcl_RemoveArpProxy, iqtcl_ClearArpProxyTable, iqtcl_SendPacketRaw methods
2.8	October 14, 2005	Added iqtcl_SendCustomCommand method
2.9	October 31, 2005	Modified iqtcl_GetCensusTable and iqtcl_GetCensusTableItem to take in column sep.
2.10	December 12, 2005	Added iqtcl_ClearStreamStats
2.11	January 27, 2006	Tcl API Version 2.2 specification Added iqtcl_GetPortCounterByIndex Added iqtcl_GetPortCounterTableItem Fixed incorrect $portvalue constants in iqtcl_StartStimulus documentation Added user defined separator capability toPortCounter strings Added ability to not loop in iqtcl_StartStimulus documentation Changed portvalue constants in PortCounter methods for consistency
2.12	February 16, 2006	Fixed case mismatches in function documentation Added iqtcl_SendTargetSyslogMsg

2007-2008	Date	Description
2.13	July 05, 2007	Version 2.7 Specification Added QAM Methods.
2.14	September 18, 2007	Added name=value census retrieval methods iqtcl_GetCensusTableItemTagged iqtcl_GetQAMTableItemTagged
2.15	January 7, 2008	Added name=value census retrieval methods: iqtcl_GetProgramTableTagged iqtcl_GetPidTableTagged Added MAC retrieval Method: iqtcl_GetMACAddress
2.16	March 27, 2008	Added new functions and documentation: General Methods General Methods: iqtcl_GetTotalAvailableMemory QAM Methods QAM Methods: iqtcl_QAMGetActiveChannel, iqtcl_QAMGetStreamID Record Methods Record Methods: iqtcl_SaveUploadRecordToFile, iqtcl_IsHostReadyForDownload, iqtcl_IsRecordDownloadComplete iqtcl_SetTriggerPos, iqtcl_AddTriggerCondition, iqtcl_RemoveTriggerCondition, iqtcl_GetTriggerStatus iqtcl_IsRecordTriggered, iqtcl_StartRecord_TriggerPos

2009	Date	Description
2.17	March 30, 2009	Synch'd functions and documentation with current firmware
2.18	August 13, 2009	Corrected functions documentation: Connection Methods Connection Methods: iqtcl_SaveSessionData, iqtcl_ResumeSession, iqtcl_RetrieveSessionData
2.19	August 19, 2009	Added Data Structure Documentation: tIXF18103CTRS, tNemoCtrs, tTritenCtrs
2.20	August 24, 2009	Corrected data structure: tTUNERSTATS, removed structures tCENCAP, tCENSUSLOOKUP

Chapter 4

Installation Procedure

The IQ Tcl-API has support for Windows, Linux, and Solaris. All platforms require a Tcl interpreter. IneoQuest requires Active State's Active Tcl, available for free download at http://activestate.com. Please download and install this Tcl runtime environment before proceeding.

For all OSes, the installation process involves the following steps:

1. Install the ActiveState Active Tcl for your OS
2. Install the IQ Tcl-API for your OS
3. Add environment variables needed to link Active Tcl and IQ Tcl-API, and to make the OS aware of the Active Tcl and IQ Tcl-API libraries. There are IQ Tcl-API library files: one that provides the fundamental routines written in C used by many IneoQuest software products (the C API library), and one that provides the routines for Tcl (the Tcl API library).

On all OSes, you can verify the installation by first checking that the Tcl-API libraries are available. Start a Tcl shell, and type the following:

```
package require iqtclapid
```

If successful, you should see the Tcl API version number in response.

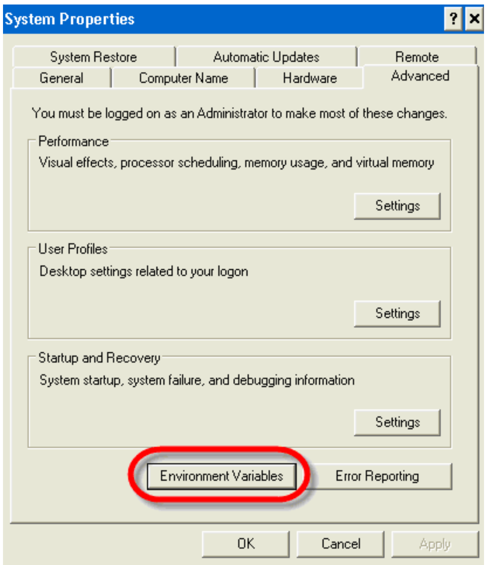
Example Tcl scripts making use of the API are in the `examples` directory wherever the Tcl-API was installed.

4.1 Installation on Windows

The IQ Tcl-API is included as part of the IQ Media Software Suite. You will need to obtain and install this software. To obtain a copy of the installer or if you have questions about installation, please email techsupport@ineoquest.com or call 1-866-GO4-INEO (1-866-464-4636).

4.1.1 Setting Windows Environment Variables

You will need to add entries to the Windows system default environment variables to link Active Tcl and IQ Tcl-API, and to make Windows aware of their libraries. The C API library goes into C:\Program Files\IneoQuest\Common (as it is used by other IneoQuest software products) and the Tcl API goes into C:\Program Files\IneoQuest\Tcl Go to My Computer - Properties - Advanced tab and click on Environment Variables.

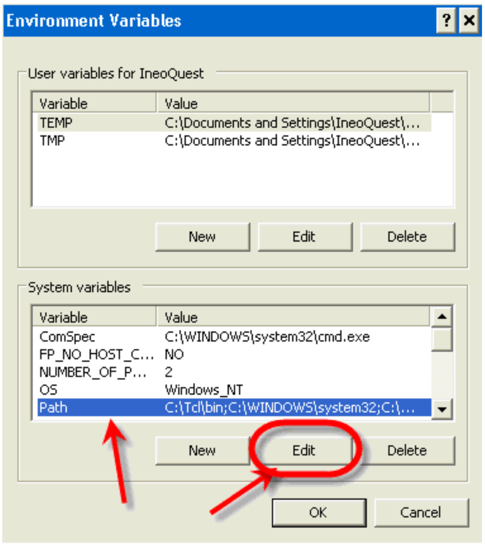


You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your logon

System startup, system failure, and debugging information



User variables for IneoQuest

Variable	Value
TEMP	C:\Documents and Settings\IneoQuest\...
TMP	C:\Documents and Settings\IneoQuest\...

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\Tcl\bin;C:\WINDOWS\system32;C:\...

The <Path environment variable needs the following at the end of the line (each value needs to be separated with a semi-colon):

```
;C:\Program Files\IneoQuest\Tcl;C:\Program Files\IneoQuest\Common;
```

The TCLLIBPATH environment variable needs to be set to

```
C:/Program\ Files/IneoQuest/Tcl
```

The slashes should be exactly as described because Active Tcl uses a Unix-like file hierarchy. The backslash is an escape character which allows Tcl to parse the space between “Program” and “Files”.

4.2 Installation on Linux

The IQ Tcl-API for Linux requires Red Hat Enterprise Linux 5, and ActiveState’s Active Tcl for Linux. Installation is performed using a self-extracting shell script which performs the needed tasks to add the ineoquest-iqapi RedHat package manager (RPM).

For the self-extracting shell installer, you can install the IQ Tcl-API via the comand:

```
./IQapi-1.0.5_Linux.sh
```

You should something like the following text:

```
bash$ ./IQapi-1.0.5_Linux.sh
```

The Tcl-API for Linux installer will install to /opt/ineoquest/iqapi . If you are not running the installer as root, the installer will not continue as superuser privileges are needed to add the software to the RPM database.

4.2.1 Linux Environment Variables

A shortcut is to edit your shell login to source the shell script /opt/ineoquest/iqapi/etc/ appropriate for your default Linux shell (currently bash and csh are supported).

The environment variables that need to be set are TCLLIBPATH, LD_LIBRARY_PATH, and PATH. Some of the settings depend on where Active Tcl is installed.

In the Bash shell, you can set these environment variables by typing:

```
export TCL_HOME=[Location where Active Tcl was installed]
export IQ_API_HOME=/opt/ineoquest/iqapi

export LD_LIBRARY_PATH=$TCL_HOME/lib:$IQ_API_HOME/lib:$LD_LIBRARY_PATH
export TCLLIBPATH=$IQ_API_HOME/lib
export PATH=$PATH:$TCL_HOME/bin
```

Similarly, in the Csh shell:

```
setenv TCL_HOME=[Location where Active Tcl was installed]
setenv IQ_API_HOME=/opt/ineoquest/iqapi

setenv LD_LIBRARY_PATH "$TCL_HOME/lib:$IQ_API_HOME/lib:$LD_LIBRARY_PATH"
setenv TCLLIBPATH $IQ_API_HOME/lib
set path=($path $TCL_HOME)
```

But note that these settings will not be persistent in another Linux shell session. By following the shortcut mentioned above, shell sessions will load these defaults.

4.3 Installation on Solaris

Two installation methods are supported under Solaris: using standard Sun PKG packages and using a self-extracting shell installer.

For those familiar with PKGs, you can install the IQ Tcl-API via the command:

```
pkgadd -d IQapi-1.0.5.pkg
```

For the self-extracting shell installer, you can install the IQ Tcl-API via the command:

```
./IQapi-1.0.5_Solaris.sh
```

You should something like the following text:

```
bash-2.03$ ./IQapi-1.0.5_Solaris.sh
Installing IQ API 1.0.5
The default directory /opt/ineoquest does not exist.
Would you like this installer to create this directory?
no
Would you rather install into your
home directory?
yes
Unarchiving activetcl.tar.gz Tcl distribution from ActiveState ... done
Unpacking IQ API for Solaris archive ... done
Direcotry /export/home/mgb/ineoquest/iqapi did not exist, creating now
Unarchiving IQ API for Solaris archive ... done
In order to properly run the IQ API Examples under a user account, you will
need to source the shell scripts in /opt/ineoquest/iqapi/etc depending on your
default shell.
 For BASH:
```

```
#!/bin/bash
. /export/home/mgb/ineoquest/iqapi/etc/bashrc
```

```
 For CSH:
```

```
source /export/home/mgb/ineoquest/iqapi/etc/csh.cshrc
```

4.3.1 Setting Solaris Environment Variables

For Solaris, the steps to set up the environment variables are similar to Linux, listed above. The exceptions come if you did not install using the PKG or installed using the shell installer but did not choose to install into the default location (rather choosing to install into your home directory). In this case, you will need to source the appropriate shell resource script.

Chapter 5

Module Documentation

5.1 Alarm Methods

Functions

- iqtcl_ClearAlarms
- iqtcl_ClearAlarmLog
- iqtcl_GetActiveAlarmFirst
- iqtcl_GetActiveAlarmNext
- iqtcl_GetActiveAlarmTable
- iqtcl_GetAlarmLogFirst
- iqtcl_GetAlarmLogNext
- iqtcl_GetAlarmLogTable

5.1.1 Detailed Description

Methods for ALARM

5.1.1.1 iqtcl_ClearAlarms

Clear any existing alarms on all streams.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ClearAlarms $connectID
```

See also:

5.1.1.2 iqtcl_ClearAlarmLog

Clears the alarm log.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ClearAlarmLog $connectID
```

See also:

5.1.1.3 iqtcl_GetActiveAlarmFirst

Returns the first entry in the Active Alarm table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structures:

- tALARMINFO
- tIPINFO
- tALARMHANDLE

Example

```
set alarmData [ iqtcl_GetActiveAlarmFirst $connectID ]
```

See also:

5.1.1.4 iqtcl_GetActiveAlarmNext

Returns the next entry in the Active Alarm table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***lastEntry*** The previous entry ID.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structures:

- tALARMINFO
- tIPINFO
- tALARMHANDLE

Example

```
set alarmData [ iqtcl_GetActiveAlarmNext $connectID $lastEntry ]
```

See also:

5.1.1.5 iqtcl_GetActiveAlarmTable

Returns a string representation of the Active Alarm table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A StringObj of the current Active Alarm table or TCL_ERROR.

Example

```
set strTable [ iqtcl_GetActiveAlarmTable $connectID ]
```

See also:

5.1.1.6 iqtcl_GetAlarmLogFirst

Returns the first entry in the Alarm Log table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structures:

- tALARMINFO
- tIPINFO
- tALARMHANDLE

Example

```
set alarmData [ iqtcl_GetAlarmLogFirst $connectID ]
```

See also:

5.1.1.7 iqtcl_GetAlarmLogNext

Returns the next entry in the Alarm Log table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***lastEntry*** The previous entry ID.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structures:

- tALARMINFO
- tIPINFO
- tALARMHANDLE

Example

```
set alarmData [ iqtcl_GetAlarmLogNext $connectID $lastEntry ]
```

See also:

5.1.1.8 iqtcl_GetAlarmLogTable

Returns a string representation of the Alarm Log table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A StringObj of the current Alarm Log table or TCL_ERROR.

Example

```
set strTable [ iqtcl_GetAlarmLogTable $connectID ]
```

See also:

5.2 Alias Methods

Functions

- iqtcl_GetAliasFirst
- iqtcl_GetAliasNext
- iqtcl_ClearAliases

5.2.1 Detailed Description

Group of alias methods

5.2.1.1 iqtcl_GetAliasFirst

Returns the first entry in the Alias table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structure:

- tALIASCONFIG

Example

```
set aliasData [ iqtcl_GetAliasFirst $connectID ]
```

See also:

5.2.1.2 iqtcl_GetAliasNext

Returns the next entry in the Alias table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***lastEntry*** The previous entry ID.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structure:

- tALIASCONFIG

Example

```
set aliasData [ iqtcl_GetAliasNext $connectID $lastEntry ]
```

See also:

5.2.1.3 iqtcl_ClearAliases

Clear aliases on probe.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ClearAliases $connectID
```

See also:

5.3 ASI Methods

Functions

- iqtcl_StartASIScan
- iqtcl_StopASIScan
- iqtcl_GetASIStatus
- iqtcl_ASILockStream
- iqtcl_ASIUnlockStream

5.3.1 Detailed Description

Methods for ASI

5.3.1.1 iqtcl_StartASIScan

Start the ASI scanning process.

Note:

The ASI scanning process will send out the locked in stream out the ASI port.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StartASIScan $connectID
```

See also:

iqtcl_StopASIScan
iqtcl_ASILockStream

5.3.1.2 iqtcl_StopASIScan

Stop the ASI scanning process.

Note:

The ASI scanning process will send out the locked in stream out the ASI port.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopASIScan $connectID
```

See also:

iqtcl_StartASIScan

5.3.1.3 iqtcl_GetASIStatus

Returns the status of the ASI process.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the data of the request or TCL_ERROR. The data array will contain the following structure:

- tASISTATUS

Example

```
set status [ iqtcl_GetASIStatus $connectID ]
```

See also:

iqtcl_ASIUnlockStream

5.3.1.4 iqtcl_ASILockStream

Locks the requested stream into the ASI process.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested stream ID.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ASILockStream $connectID $streamID
```

See also:

iqtcl_ASIUnlockStream

5.3.1.5 iqtcl_ASIUnlockStream

Removes the requested stream from the ASI process.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested stream ID.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ASIUnlockStream $connectID $streamID
```

See also:

iqtcl_ASILockStream

5.4 Capture Methods

Functions

- iqtcl_OpenCapture
- iqtcl_OpenStreamCapture
- iqtcl_CloseCapture
- iqtcl_StartCapture
- iqtcl_StopCapture
- iqtcl_UploadCapture
- iqtcl_GetCaptureStatus

5.4.1 Detailed Description

Methods for capture

5.4.1.1 iqtcl_OpenCapture

Opens a new capture session.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A 32-bit capture Handle or TCL_ERROR.

Example

```
set captureHandle [ iqtcl_OpenCapture $connectID ]
```

See also:

5.4.1.2 iqtcl_OpenStreamCapture

Opens a new capture session setting a filter to capture only packets from a requested stream ID.

Note:

The stream ID is supplied as part of the census retrieval, or can be retrieved with the iqtcl_GetStreamID API call.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The stream ID to capture.

Returns:

A 32-bit capture Handle or TCL_ERROR.

Example

```
set captureHandle [ iqtclOpenStreamCapture $connectID $streamID ]
```

See also:

iqtcl_GetStreamID

5.4.1.3 iqtcl_CloseCapture

Closes the open capture session.

Note:

Closing the capture will release all resources reserved during the iqtcl_OpenCapture method.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***handleID*** The capture Handle ID returned from the OpenCapture method.

Returns:

TCL_OK or TCL_ERROR

Example

```
 iqtcl_CloseCapture $connectID handleID
```

See also:

iqtcl_OpenCapture

5.4.1.4 iqtcl_StartCapture

Start capturing packets on the wire.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***captureHandle*** The capture handle returned from the iqtcl_OpenCapture or iqtcl_OpenStreamCapture methods.

← ***flags*** A 32-bit flags bitmask value with the following bits defined:

1. 0x1: Circular/One Shot Mode
   (a) (Value = 1) One shot mode. Capture packets until buffer is full then stop.
   (b) (Value = 0) Capture into a circular buffer.
2. 0x2 = Record Headers
   (a) (Value = 1) Capture full packet headers with data.
   (b) (Value = 0) Capture packet data bytes only (Skip encapsulation headers).

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StartCapture $connectID $captureHandle $flags
```

See also:

5.4.1.5 iqtcl_StopCapture

Stop capturing packets on the wire.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***captureHandle*** The capture handle returned from the iqtcl_OpenCapture or iqtcl_OpenStreamCapture methods.

← ***upload*** A boolean indicating whether or not the captured packets should be uploaded from the FPGA after stopping the capture.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopCapture $connectID $captureHandle
```

See also:

5.4.1.6 iqtcl_UploadCapture

Upload the captured packet information from the target.

Note:

The capture upload will be done via a TCP connection. The requesting application should open a socket and listen. This local socket port is passed to the target as part of this API call. The target will connect to this listening socket and start sending captured packet information. When the target has completed sending the data it will close the connection. The packet data arrives in the following format:

- 16 bit tag ID: The tag ID is always 0xD.
- 16 bit data size: The data size is the total number of data bytes to follow, i.e. the number of bytes of packet data + 8 bytes for the timestamp.
- 64 bit timestamp: The timestamp has a 10ns resolution.
- n bits of packet data: The packet data may or may not include the packet encapsulation depending on what flags were passed into the iqtcl_StartCapture API method.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***captureHandle*** The capture handle returned from the iqtcl_OpenCapture or iqtcl_-OpenStreamCapture methods.

← ***tcport*** The socket port the target should connect to to start sending packets.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_UploadCapture $connectID $captureHandle $tcport
```

See also:

5.4.1.7 iqtcl_GetCaptureStatus

Get the status of the capture process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***captureHandle*** The capture Handle ID returned from the OpenCapture method.

Returns:

A Tcl_List structure or TCL_ERROR. The first item in the list is the Status flag defined as:

- 0 = INACTIVE - capture has inactive and ready to upload data
- 1 = ACTIVE - capture process is currently capturing packets
- 2 = DOWNLOADING - capture process is getting data from internal buffers
- 3 = UPLOADING - capture process is uploading data to host

The second item in the list is the number of packets in the capture buffer.

Example

```
set statusList [ iqtcl_GetCaptureStatus $connectID $captureHandle ]
```

See also:

iqtcl_OpenCapture

5.5 Census Methods

Functions

- iqtcl_GetCensusFirst
- iqtcl_GetCensusNext
- iqtcl_GetCensusByID
- iqtcl_GetNextCensusID
- iqtcl_GetStreamID
- iqtcl_ClearCensus
- iqtcl_ClearStream
- iqtcl_ClearStreamStats
- iqtcl_GetCensusTableItem
- iqtcl_GetCensusTableItemTagged
- iqtcl_GetQAMTableItem
- iqtcl_GetQAMTableItemTagged
- iqtcl_GetQAMTable
- iqtcl_GetCensusTable
- iqtcl_GetProgramTableTagged
- iqtcl_GetPidTableTagged

5.5.1 Detailed Description

Methods to get census info

5.5.1.1 iqtcl_GetCensusFirst

Returns the census first entry in the Census table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain a tCENTRY structure followed by zero or more of the following structures:

- tIPINFO
- tENCAP
- tMPEG2INFO
- tMPEG2PROGRAM
- tMPEG2PID
- tRTPINFO
- tMDIINFO
- tSTREAMSTATS
- tMTSPSTATS
- tRTPSTATS

Example

```
set censusData [ iqtcl_GetCensusFirst $connectID ]
```

See also:

5.5.1.2 iqtcl_GetCensusNext

Returns the next entry in the Census table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***previousID*** The previous entry ID.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain a tCENTRY structure followed by zero or more of the following structures:

- tIPINFO
- tENCAP
- tMPEG2INFO
- tMPEG2PROGRAM
- tMPEG2PID
- tRTPINFO
- tMDIINFO
- tSTREAMSTATS
- tMTSPSTATS
- tRTPSTATS

Example

```
set censusData [ iqtcl_GetCensusNext $connectID $previousID ]
```

See also:

5.5.1.3 iqtcl_GetCensusByID

Returns the census information of the requested stream ID.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The ID of the requested stream

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain a tCENTRY structure followed by zero or more of the following structures:

- tIPINFO
- tENCAP
- tMPEG2INFO
- tMPEG2PROGRAM
- tMPEG2PID
- tRTPINFO
- tMDIINFO
- tSTREAMSTATS
- tMTSPSTATS
- tRTPSTATS

Example

```
set censusData [ iqtcl_GetCensusByID $connectID $streamID ]
```

See also:

5.5.1.4 iqtcl_GetNextCensusID

Returns the next census information of the previous ID.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***lastID*** The ID of the previous requested stream

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain a tCENTRY structure followed by zero or more of the following structures:

- tIPINFO
- tENCAP
- tMPEG2INFO
- tMPEG2PROGRAM
- tMPEG2PID
- tRTPINFO
- tMDIINFO
- tSTREAMSTATS
- tMTSPSTATS
- tRTPSTATS

Example

```
set censusData [ iqtcl_GetNextCensusID $connectID $lastID ]
```

See also:

5.5.1.5 iqtcl_GetStreamID

This function will return the ID of the stream matching the passed in IP address tuple.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***srcIP*** An integer representing the source IP of desired stream.

← ***dstIP*** An integer representing the destination IP of desired stream

← ***srcPort*** An integer representing the source port of desired stream

← ***dstPort*** An integer representing the destination port of desired stream

← ***TargetPort*** Used to distinguish between the same tuple on multiple physical ports. If TargetPort is 0, match flow using main port 1; if non-zero, flow must be on port 2.

Returns:

The streamID of matching stream or TCL_ERROR.

Example

To get the stream ID with source IP 224.0.1.2, port 1024 to destination IP 10.1.0.1, port 2048 on probe port 1.

```
set srcIP 0xE0000102
set dstIP 0x0A010001
set srcPort 1024
set dstPort 2048
set streamID [ iqtcl_GetStreamID $srcIP $dstIP $srcPort $dstPort 0 ]
```

See also:

5.5.1.6 iqtcl_ClearCensus

Clears the current census table.

Note:

This method will clear all detected streams from the census table. If any streams are still active, they will be re-acquired into the census table, but all previous statistical information will be cleared.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ClearCensus $connectID
```

See also:

5.5.1.7 iqtcl_ClearStream

Clears a single stream from the current census table.

Note:

This method will clear a single stream from the census table. If the stream is still active, it will be re-acquired into the census table, but all previous statistical information will be cleared.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The stream ID

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ClearStream $connectID $streamID
```

See also:

5.5.1.8 iqtcl_ClearStreamStats

Clears a single stream or the entire census statistics.

Note:

This method will clear the statistics of a single stream or the entire census table.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The stream ID or 0x0 to clear all streams.

← ***flags*** A flags value indicating which stats to clear. Valid values include an OR of any of the following:

1. 0x00000001 - clear MDI information
2. 0x00000002 - clear Stream rate statistics
3. 0x00000004 - clear RTP statistics
4. 0x00000008 - clear MTSP statistics
5. 0x00000010 - clear MPEG Pid Statistics
6. 0xFFFFFFFF - clear ALL statistics

Returns:

TCL_OK or TCL_ERROR

Example

```
 iqtcl_ClearStreamStats $connectID $streamID 0xffffffff
```

See also:

5.5.1.9 iqtcl_GetCensusTableItem

Returns a string representation of the requested flow's census information.

Note:

The censusID can be retrieved by a call to iqtcl_GetNextCensusID.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***censusID*** The requested census ID.

← ***separator*** The desired column separator string.

Returns:

A StringObj of the requested census table stream entry or TCL_ERROR.

Example

```
set strItem [ iqtcl_GetCensusTableItem $connectID $censusID " " ]
```

See also:

5.5.1.10 iqtcl_GetCensusTableItemTagged

Returns a tagged string representation of the requested flow's census information.

Note:

The tagged string is defined as `<name>=<value>;<name>=<value>;` where `<name>` and `<value>` are the same strings returned by a call to iqtcl_GetCensusTable. The censusID can be retrieved by a call to iqtcl_GetNextCensusID.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***censusID*** The requested censusID.

Returns:

A StringObj of the requested census table flow entry or TCL_ERROR.

Example

```
set strItem [ iqtcl_GetCensusTableItemTagged $connectID $censusID ]
```

See also:

5.5.1.11 iqtcl_GetQAMTableItem

Returns a tagged string representation of the requested flow's census information.

Note:

The tagged string is defined as `<name>=<value>;<name>=<value>;` where `<name>` and `<value>` are the same strings returned by a call to iqtcl_GetQAMTable.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested censusID.

Returns:

A StringObj of the requested census table flow entry or TCL_ERROR.

Example

```
set strItem [ iqtcl_GetQAMTableItemTagged $connectID $streamID ]
```

See also:

5.5.1.12 iqtcl_GetQAMTableItemTagged

Returns a tagged string representation of the requested flow's census information.

Note:

The tagged string is defined as `>name<=>value<;>name<=>value<; where >name< and >value< are the same strings returned by a call to iqtcl_GetQAMTable.`

Parameters:

← ***connectID*** `The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.`

← ***streamID*** `The requested censusID.`

Returns:

`A StringObj of the requested census table flow entry or TCL_ERROR.`

Example

```
set strItem [ iqtcl_GetQAMTableItemTagged $connectID $streamID ]
```

See also:

5.5.1.13 iqtcl_GetQAMTable

Returns a string representation of the current census table for a QAM device.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***sepStr*** The desired column separator string.

Returns:

A StringObj of the current census table or TCL_ERROR.

Example

```
set strTable [ iqtcl_GetQAMTable $connectID "\t" ]
```

See also:

5.5.1.14 iqtcl_GetCensusTable

Returns a string representation of the current census table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***sepStr*** The desired column separator string.

Returns:

A StringObj of the current census table or TCL_ERROR.

Example

```
set strTable [ iqtcl_GetCensusTable $connectID "\t" ]
```

See also:

5.5.1.15 iqtcl_GetProgramTableTagged

Returns a tagged string representation of the requested flow's program information.

Note:

The tagged string is defined as name/value pairs separated by ';'. Each program information is separated by '::'. An example string would be 'Channel=1;Name=Program1;'Channel=2;Name=Prog2;''

The following items are returned for each channel:

- Channel - channel number of program
- Name - the name of this program defined in the TS.
- Alias - the Ineoquest specific alias defined for this program
- DevRef - the Ineoquest specific subreference defined for this program
- Provider - the provider string for this program defined in the SDT
- NumPids - the number of pids in this program
- AlarmStatus - the alarm status of this program

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested censusID.

Returns:

A StringObj of the requested program information or TCL_ERROR.

Example

```
set strItem [ iqtcl_GetProgramTableTagged $connectID $streamID ]
```

See also:

5.5.1.16 iqtcl_GetPidTableTagged

Returns a tagged string representation of the requested program's pid information.

Note:

The tagged string is defined as name/value pairs separated by ';'. Each pid information is separated by '::'. An example string would be Channel=1;Name=Program1;Channel=2;Name=Prog2;

The following items are returned for each channel:

- PID - the PID number
- Type - the type of pid this is (audio/video/pcr).
- CurrBR - current pid bitrate
- MinBR - minimum pid bitrate
- MaxBR - maximum pid bitrate
- AvgBR - average pid bitrate
- OutageCount - number of outages seen for this pid
- OutageSec - duration of outage for this pid
- CurrLoss - loss count of this pid over the last 15 minutes
- TotLoss - total loss count of this pid since discovery
- Alarm_Status - alarm status of this pid

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested censusID.

← ***progChannel*** The requested programChannel;

Returns:

A StringObj of the requested program information or TCL_ERROR.

Example

```
set strItem [ iqtcl_GetProgramTableTagged $connectID $streamID $progChannel ]
```

See also:

5.6 Connection Methods

Functions

- iqtcl_ConnectTo
- iqtcl_OpenConnection
- iqtcl_OpenConnectionWithMsgSize
- iqtcl_CloseConnection
- iqtcl_IsConnected
- iqtcl_CloseSession
- iqtcl_ResumeSession
- iqtcl_CloseOldConnections
- iqtcl_SaveSessionData
- iqtcl_RetrieveSessionData

5.6.1 Detailed Description

Methods for setting up and maintaining connections and sessions.

5.6.1.1 iqtcl_ConnectTo

Opens a new connection to a target.

Note:

This method takes a string representation of the target's IP address. For numerical representations use the iqtcl_OpenConnection method.

Parameters:

← ***char**** A string representation of the target's IP address.

Returns:

A connection ID used in subsequent communications with the target or TCL_ERROR.

Example

```
set connectionID [ iqtcl_ConnectTo ``192.168.1.1'' ]
```

See also:

iqtcl_OpenConnection

5.6.1.2 iqtcl_OpenConnection

Opens a new connection to a target.

Note:

This method takes a numerical representation of the target's IP address. For string representations use the iqtcl_ConnectTo method.

Parameters:

← ***NUM*** A numerical representation of the target's IP address.

← ***FLAG*** A boolean flag indicating if the IP address is in network order.

Returns:

A connection ID used in subsequent communications with the target or TCL_ERROR.

Example

To connect to 192.168.1.1 with the ip in network order:

```
set connectionID [ iqtcl_OpenConnection 0xc0a80101 1 ]
```

See also:

iqtcl_ConnectTo

5.6.1.3 iqtcl_OpenConnectionWithMsgSize

Opens a new connection to a target and sets the preferred comm message size.

Note:

This method takes a numerical representation of the target's IP address. For string representations use the iqtcl_ConnectTo method.

Parameters:

← ***NUM*** A numerical representation of the target's IP address.

← ***MSG_SIZE*** A number specifying the preferred message size for communication

← ***FLAG*** A boolean flag indicating if the IP address is in network order.

Returns:

A connection ID used in subsequent communications with the target or TCL_ERROR.

Example

To connect to 192.168.1.1 with the ip in network order:

```
set connectionID [ iqtcl_OpenConnection 0xc0a80101 $msg_size 1 ]
```

See also:

iqtcl_ConnectTo

5.6.1.4 iqtcl_CloseConnection

Closes open connection. Closes a currently open connection.

Note:

This method will release all resources reserved by the current session before closing the connection.

Parameters:

← ***connectionID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_CloseConnection $connectionID
```

See also:

5.6.1.5 iqtcl_IsConnected

Specifies whether the session is connected to a target.

Note:

Parameters:

← ***connectionID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A BooleanObj indicating whether the session is connected or not or TCL_ERROR.

Example

```
set state [ iqtcl_IsConnected $connectionID ]
```

See also:

5.6.1.6 iqtcl_CloseSession

Closes the current connections but does not release any resources currently being used.

Note:

This method is used in conjunction with the iqtcl_ResumeSession method to reconnect to a target without changing the state of the target. Any user-defined state information can be stored on the target with the iqtcl_SaveSessionData method.

Parameters:

← ***connectionID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_CloseSession $connectionID
```

See also:

iqtcl_ResumeSession iqtcl_SaveSessionData

5.6.1.7 iqtcl_ResumeSession

Restores a previous session.

Note:

This method is used in conjunction with the iqtcl_CloseSession method to reconnect to a target without changing the state of the target. Any resources that were owned by the previous session will now be owned by the current session. Any saved user-defined state information can be retrieved from the target with the iqtcl_RetrieveSessionData method.

Parameters:

← ***connectionID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***sessionID*** The sessionID returned from the iqtcl_ RetrieveSessionData method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ResumeSession $connectionID $sessionID
```

See also:

iqtcl_CloseSession iqtcl_RetrieveSessionData

5.6.1.8 iqtcl_CloseOldConnections

Releases any resources and closes connections to any sessions other than the current one.

Note:

This method will release all resources reserved by any sessions other than the current one.

Parameters:

← ***connectionID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_CloseOldConnections $connectionID
```

See also:

5.6.1.9 iqtcl_SaveSessionData

Saves user session data on the target.

Note:

This method is used in conjunction with the iqtcl_ RetrieveSessionData method to store user session data on the target and retrieve it when a session is resumed.

Parameters:

← ***connectionID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***data*** A ByteArrayObj containing the data to save to the target.

← ***datasize*** The size of the ByteArrayObj data

← ***mode*** Mode for saving session data, active (0) or passive (1)

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_SaveSessionData $connectionID $data $datasize $mode
```

See also:

iqtcl_RetrieveSessionData

5.6.1.10 iqtcl_RetrieveSessionData

Retrieves the user session data saved on the target during a previous iqtcl_SaveSessionData call.

Note:

This method is used in conjunction with the iqtcl_SaveSessionData method to store user session data on the target.

Parameters:

← ***connectionID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***resumeHandle*** A placeholder variable in which the resume handle will be returned. This handle is used in the iqtcl_ResumeSession method.

← ***mode*** Mode for saving session data, active (0) or passive (1)

Returns:

A byte array of the user data stored on the target during the last iqtcl_SaveSessionData call or TCL_ERROR.

Example

```
 set data [ iqtcl_RetrieveSessionData $connectionID $resumeHandle $mode]
```

See also:

iqtcl_SetSessionData

5.7 General Methods

Functions

- iqtcl_SetDefaultTimeout
- iqtcl_GetLastError
- iqtcl_GetAPIVersionString
- iqtcl_GetMACAddress
- iqtcl_GetCoreMode
- iqtcl_SetCoreMode
- iqtcl_GetTargetType
- iqtcl_GetFirmwareMode
- iqtcl_GetTargetInfo
- iqtcl_GetTotalAvailableMemory
- iqtcl_SendSyslogMsg
- iqtcl_SendTargetSyslogMsg
- iqtcl_SendCustomCommand

5.7.1 Detailed Description

Group of generally useful methods

5.7.1.1 iqtcl_SetDefaultTimeout

Sets the default socket timeout for the API.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***timeout*** The new default timeout in seconds.

Returns:

TCL_OK or TCL_ERROR

Example

To set the default timeout to 10 seconds:

```
iqtcl_SetDefaultTimeout $connectID 10
```

See also:

5.7.1.2 iqtcl_GetLastError

Returns the last error stored in the API.

Note:

This method can be called following a failed method call to get the reason for failure.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A StringObj variable of the current version or TCL_ERROR.

Example

```
set error [iqtcl_GetLastError $connectID ]
```

See also:

iqtcl_ConnectTo
iqtcl_OpenConnection

5.7.1.3 iqtcl_GetAPIVersionString

Returns the version string of the underlying C-API library.

Note:

None.

Parameters:

None.

Returns:

A StringObj variable of the current version or TCL_ERROR.

Example

```
set strVersion [ iqtcl_GetAPIVersionString ]
```

See also:

5.7.1.4 iqtcl_GetMACAddress

Returns the MAC address values for the requested port.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***port*** The requested port. Value should be one of :

- Management Port = 0x02000000
- Port1 = 0xA0000000
- Port2/TAP/RVL = 0xA0800000

Returns:

A StringObj of the requested MAC address or TCL_ERROR.

Example

```
set strMAC [ iqtcl_GetMacAddress $connectID 0x02000000 ]
```

See also:

5.7.1.5 iqtcl_GetCoreMode

Gets the target into the requested analysis mode.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

mode The target mode, defined as:

- 0 = Census Mode
- 1 = Analysis Mode

Example

To check if the target into Census Mode:

```
set state [ iqtcl_GetCoreMode $connectID ]
```

See also:

iqtcl_SetCoreMode

5.7.1.6 iqtcl_SetCoreMode

Sets the target into the requested analysis mode.

Note:

If the target is place into Analysis Mode, new streams will not be detected until it is placed back into Census Mode. Analysis Mode is used when analyzing a particular stream for errors and recording a particular stream.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***mode*** The requested target mode, defined as:

- 0 = Census Mode
- 1 = Analysis Mode

Returns:

TCL_OK or TCL_ERROR

Example

To set the target into Census Mode:

```
iqtcl_SetCoreMode $connectID 0
```

See also:

iqtcl_GetCoreMode

5.7.1.7 iqtcl_GetTargetType

Returns the type of target you are connected to.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

The current target type or TCL_ERROR. Target type is defined as:
1 = SingulusG1
2 = SingulusG1-T
3 = SingulusG10
4 = Argus
5 = Probe

Example

```
set ttype [ iqtcl_GetTargeType $connectID ]
```

See also:

5.7.1.8 iqtcl_GetFirmwareMode

Returns the current mode the target firmware is running in.

Note:

The firmware mode returned is set through the target's HTML interface.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A target's current mode or TCL_ERROR. Target mode is defined as:

- 0 = IQController Mode
- 1 = Analysis Mode
- 2 = Stimulus Mode
- 3 = Maintenance Mode

Example

```
set mode [iqtcl_GetFirmwareMode $connectID ]
```

See also:

iqtcl_ConnectTo
iqtcl_OpenConnection

5.7.1.9 iqtcl_GetTargetInfo

Returns a byte array of the target's current information.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A byte array of target information or TCL_ERROR. The target information is defined by the tTARGETINFO structure.

Example

```
set info [ iqtcl_GetTargetInfo $connectID ]
```

See also:

iqtcl_ConnectTo
iqtcl_OpenConnection

5.7.1.10 iqtcl_GetTotalAvailableMemory

Gets total available memory from probe.

Note:

None.

Parameters:

connectID The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

Size of total available memory on probe (in bytes)

Example

```
set availMem [ iqtcl_GetTotalAvailableMemory $connectID ]
```

See also:

Author:

mgb

5.7.1.11 iqtcl_SendSyslogMsg

Send a message to a syslog server.

Note:

Parameters:

← ***IpAddr*** The IP address string of the syslog server.

← ***MsgStr*** The message string to send

Returns:

TCL_OK or TCL_ERROR.

Example

```
iqtcl_SendSyslogMsg 192.168.1.1 "This is a test"
```

See also:

5.7.1.12 iqtcl_SendTargetSyslogMsg

Tell the connected target to send a syslog message to its configured syslog server with the proper headers.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***messageStr*** The message string to send.

Returns:

TCL_OK or TCL_ERROR.

Example

```
iqtcl_SendTargetSyslogMsg $connectID "This is the start of a test"
```

See also:

5.7.1.13 iqtcl_SendCustomCommand

Send a custom string command to the target.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***command*** The char∗ command

Returns:

TCL_OK or TCL_ERROR

Example

```
See also:
```

5.8 ARP Proxy Methods

Functions

- iqtcl_AddArpProxy
- iqtcl_RemoveArpProxy
- iqtcl_ClearArpProxyTable
- iqtcl_AddArpProxyEx
- iqtcl_RemoveArpProxyEx

5.8.1 Detailed Description

Group of ARP Proxy Methods methods

5.8.1.1 iqtcl_AddArpProxy

Add an entry to the proxy arp table.

Note:

None

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***destinationMAC*** The destination MAC sent as a byte array.

← ***VlanID*** The VLAN ID if used or 0xFFFFFFFF if not used.

← ***SrcIPstr*** The source IP sent as a string.

Returns:

TCL_ERROR or TCL_OK

Example

```
set arpMac [binary format H* 0008d0123456]
iqtcl_AddArpProxy $connectID $arpMac 0xFFFFFFFF "192.168.1.1"
```

See also:

iqtcl_RemoveArpProxy

5.8.1.2 iqtcl_RemoveArpProxy

Remove an entry from the proxy arp table.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***destinationMAC*** The destination MAC sent as a byte array.

← ***VlanID*** The VLAN ID if used or 0xFFFFFFFF if not used.

← ***SrcIPstr*** The source IP sent as a string.

Returns:

TCL_ERROR or TCL_OK

Example

```
set arpMac [binary format H* 0008d0123456]
iqtcl_RemoveArpProxy $connectID $arpMac 0xFFFFFFFF "192.168.1.1"
```

See also:

iqtcl_AddArpProxy

5.8.1.3 iqtcl_ClearArpProxyTable

Clear the proxy ARP table.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_ERROR or TCL_OK

Example

```
 iqtcl_ClearArpProxyTable $connectID
```

See also:

5.8.1.4 iqtcl_AddArpProxyEx

Add passed in data to the proxy ARP table.

Note:

Not Implemented.

5.8.1.5 iqtcl_RemoveArpProxyEx

Remove passed in data from the proxy ARP table.

Note:

Not Implemented.

5.9 IGMP Methods

Functions

- iqtcl_GetIGMPStatus
- iqtcl_GetIGMPFirst
- iqtcl_GetIGMPNext
- iqtcl_IGMPJoin
- iqtcl_IGMPLeave
- iqtcl_StartIGMPLoop
- iqtcl_StopIGMPLoop
- iqtcl_ClearIGMP

5.9.1 Detailed Description

Methods for IGMP

5.9.1.1 iqtcl_GetIGMPStatus

Returns the status of the IGMP process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the IGMP status or TCL_ERROR. The data array will contain the following structures:

- tIGMPSTATUS
- tIGMPGROUPS

Example

```
set igmpstatusResult [ iqtcl_GetIGMPStatus $connectID]
```

See also:

5.9.1.2 iqtcl_GetIGMPFirst

Returns the first entry in the IGMP table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structures:

- tIGMPSTATS
- tNAMETAG
- tIGMPEVENT

Example

```
set alarmData [ iqtcl_GetIGMPFirst $connectID]
```

See also:

5.9.1.3 iqtcl_GetIGMPNext

Returns the next entry in IGMP table.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***previousID*** The previous entry ID.

Returns:

A ByteArrayObj containing the data of the requested entry or TCL_ERROR. The data array will contain the following structures:

- tIGMPSTATS
- tNAMETAG
- tIGMPEVENT

Example

```
set alarmData [ iqtcl_GetIGMPNext $connectID $lastEntry]
```

See also:

5.9.1.4 iqtcl_IGMPJoin

Join the passed in multicast address.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***addr*** The integer representation of the multicast address to join.

Returns:

TCL_OK or TCL_ERROR

Example

To join address 224.1.2.3:

```
set address 0xE0010203
iqtcl_IGMPJoin $address
```

See also:

iqtcl_IGMPLeave

5.9.1.5 iqtcl_IGMPLeave

Leave the passed in multicast address.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***addr*** The integer representation of the multicast address to join.

Returns:

TCL_OK or TCL_ERROR

Example

To leave address 224.1.2.3:

```
set address 0xE0010203
iqtcl_IGMPLeave $address
```

See also:

iqtcl_IGMPJoin

5.9.1.6 iqtcl_StartIGMPLoop

Start the IGMP automatic loop.

Note:

The IGMP loop will automatic ally join and leave the defined multicast addresses as defined in the IGMP configuration.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StartIGMPLoop
```

See also:

iqtcl_StopIGMPLoop

5.9.1.7 iqtcl_StopIGMPLoop

Stop the IGMP automatic loop.

Note:

The IGMP loop will automatic ally join and leave the defined multicast addresses as defined in the IGMP configuration.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopIGMPLoop
```

See also:

iqtcl_StartIGMPLoop

5.9.1.8 iqtcl_ClearIGMP

Clears the current IGMP statistical information.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR.

Example

```
iqtcl_ClearIGMp $connectID
```

See also:

5.10 STB Methods

Functions

- iqtcl_OpenSTBSession
- iqtcl_CloseSTBSession
- iqtcl_StartSTBSession
- iqtcl_StopSTBSession
- iqtcl_ClearSTBDefines
- iqtcl_DefineNewSTB
- iqtcl_GetSTBDataTable

5.10.1 Detailed Description

Methods for STB

5.10.1.1 iqtcl_OpenSTBSession

Opens a STB Mode session on the target.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A 32-bit STB handle or TCL_ERROR.

Example

```
set stbHandle [ iqtclOpenSTBSession $connectID]
```

See also:

5.10.1.2 iqtcl_CloseSTBSession

Close a STB Mode session on the target.

Note:

Closing the STB session will release all resources reserved during the iqtcl_OpenSTBSession API method.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stbID*** The STB handle returned from the iqtcl_OpenSTBSession API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
 iqtcl_CloseSTBSession $connectID $stbHandle
```

See also:

iqtcl_OpenSTBSession

5.10.1.3 iqtcl_StartSTBSession

Start the STB Session running.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stbID*** The STB handle returned from the iqtcl_OpenSTBSession API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtclStartSTBSession $connectID $stbHandle
```

See also:

5.10.1.4 iqtcl_StopSTBSession

Stop a running STB session.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stbHandle*** The STB handle returned from the iqtcl_OpenSTBSession API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtclStopSTBSession $connectID $stbHandle
```

See also:

5.10.1.5 iqtcl_ClearSTBDefines

Clear any previously defined STB definitions.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stbID*** The STB handle returned from the iqtcl_OpenSTBSession API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ClearSTBDefines $connectID $stbHandle
```

See also:

5.10.1.6 iqtcl_DefineNewSTB

Define a new virtual set-top box on the target.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stbHandle*** The STB Handle returned from the iqtcl_OpenSTBSession API call.

← ***macVal*** The MAC address of the virtual set-top box.

← ***vlanField*** The VLAN field of the virtual set-top box OR the value 0xFFFFFFFF if the settop box will not use VLANs.

← ***ipAddrStr*** The IP address string (in dot notation) of the virtual set-top box.

← ***index*** The starting index of groups to join. This is an offset index into the range of groups defined by the first index and the number of groups.

← ***numGroups*** The number of consecutive groups in the target's alias table that will be joined (starting from the first index).

← ***index*** The first index defines the beginning of the groups for the virtual set-top box. It is an index into the group of multicast aliases defined on the target.

← ***lenJoinDuration*** The length of time each group will remain joined defined in microseconds.

← ***lenJoinTransition*** The length of time between group joins defined in microseconds.

Returns:

A 32-bit virtual STB handle or TCL_ERROR.

Example

```
 set macVal [binary format H* 0008d0123456]


 set boxHandle [ iqtclDefineNewSTB $connectID $stbHandle $macVal 0xFFFFFFFF
192.168.10.10 0 5 0 7500 2500]
```

See also:

5.10.1.7 iqtcl_GetSTBDataTable

Returns a string representation of the current STB table.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

Example

See also:

5.11 License Methods

Functions

- iqtcl_IsLicenseValid
- iqtcl_GetLicenseCount

5.11.1 Detailed Description

Methods for LICENSE

5.11.1.1 iqtcl_IsLicenseValid

Does the target have a valid license for the requested license type.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***licenseType*** The requested license type can be one of the following:

- 1. = Stimulus License
- 2. = TSReader License

Returns:

TCL_OK or TCL_ERROR

Example

To check if the target has a valid TSReader license:

```
iqtcl_IsLicenseValid $connectID 2
```

See also:

5.11.1.2 iqtcl_GetLicenseCount

Returns the number of uses left on a given license type.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***licenseType*** The requested license type can be one of the following:

- 1. = Stimulus License
- 2. = TSReader License

Returns:

The number of uses left on the license or TCL_ERROR

Example

To get the number of uses remaining on the stimulus license:

```
set licCount [ iqtcl_GetLicenseCount $connectID 1 ]
```

See also:

5.12 MDI Methods

Functions

- iqtcl_StartMDIScan
- iqtcl_StopMDIScan
- iqtcl_GetMDIStatus
- iqtcl_MDILockStream
- iqtcl_MDIUnlockStream

5.12.1 Detailed Description

Methods for MDI

5.12.1.1 iqtcl_StartMDIScan

Start the MDI scanning process.

Note:

The MDI scanning process will calculate the current MDI for any streams that are locked into the MDI process.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StartMDIScan $connectID
```

See also:

iqtcl_StopMDIScan
iqtcl_MDILockStream

5.12.1.2 iqtcl_StopMDIScan

Stop the MDI scanning process.

Note:

The MDI scanning process will calculate the current MDI for any streams that are locked into the MDI process.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopMDIScan $connectID
```

See also:

iqtcl_StartMDIScan
iqtcl_MDILockStream

5.12.1.3 iqtcl_GetMDIStatus

Returns the status of the MDI process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

The current status of the MDI process. The status is defined as follows:

- 0 = MDI process is inactive
- 1 = MDI process is active

Example

```
set status [ iqtcl_GetMDIStatus $connectID ]
```

See also:

iqtcl_StartMDIScan
iqtcl_MDILockStream

5.12.1.4 iqtcl_MDILockStream

Locks the requested stream into the MDI process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested stream ID.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_MDILockStream $connectID $streamID
```

See also:

iqtcl_MDIUnlockStream
iqtcl_GetStreamID

5.12.1.5 iqtcl_MDIUnlockStream

Removes the requested stream from the MDI process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested stream ID.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_MDIUnlockStream $connectID $streamID
```

See also:

iqtcl_MDILockStream

5.13 QAM Methods

Functions

- iqtcl_QAMGetState
- iqtcl_QAMGetActiveChannel
- iqtcl_QAMGetStreamID
- iqtcl_QAMGenerateAliasesFromLearn
- iqtcl_QAMStartStream
- iqtcl_QAMStartScan
- iqtcl_IsQAMScanning
- iqtcl_QAMStartLearn
- iqtcl_QAMStartOp
- iqtcl_QAMTuneAndStreamBySTBChanName
- iqtcl_QAMTuneAndStreamBySTBChanNumber
- iqtcl_QAMStopStream
- iqtcl_QAMStopScan
- iqtcl_QAMStopLearn
- iqtcl_QAMStopOp

5.13.1 Detailed Description

Methods for QAM

5.13.1.1 iqtcl_QAMGetState

Returns the current state of the QAM system as an enumerated value.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***chanNum*** An integer defining the channel number you want to get the state of or 0 for current active channel.

Returns:

TCL_ERROR or an enumerated value for QAM state. Enumerations are defined as:

0	IDLE
3	LEARNING
4	TUNED (the QAM system is on the channel but not actively taking measurements)
5	STREAMING
10	NO SIGNAL ERROR
11	GENERIC ERROR

Example

```
set state [ iqtcl_QAMGetState $connectID 0 ]
```

See also:

5.13.1.2 iqtcl_QAMGetActiveChannel

Get active channel on QAM Cricket.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

Example

```
set actChannel [ iqtcl_QAMGetActiveChannel $connectID ]
```

See also:

5.13.1.3 iqtcl_QAMGetStreamID

Get the stream ID for a channel.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***chanNum*** The channel number

Returns:

Example

```
set QAMstreamId [ iqtcl_QAMGetStreamID $connectID $chan ]
```

See also:

5.13.1.4 iqtcl_QAMGenerateAliasesFromLearn

Creates alias information on the target based on channels found during the learning phase.

Note:

This method must be called after a learning phase has completed.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_ERROR or TCL_OK

Example

```
 iqtcl_QAMGenerateAliasesFromLearn $connectID
```

See also:

5.13.1.5 iqtcl_QAMStartStream

Tunes in to a defined channel and monitors/measures that channel.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***chanNum*** The channel number to monitor.

Returns:

TCL_ERROR or TCL_OK

Example

```
iqtcl_QAMStartStream $connectID 4
```

See also:

5.13.1.6 iqtcl_QAMStartScan

Starts a scan operation on the QAM System.

A scan will tune to a particular channel for a specified length of time and take measurements on that channel. It will then tune in the next channel DEFINED AS AN ALIAS and repeat the measurements on that channel. It will scan over all channels defined in the Alias table.

Note:

The scan will only operate on channels that have been defined with an alias.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***chanNum*** The channel number to start the scan on (or 0 for current active channel).

← ***lenTime*** The lenth of time in seconds to scan the channel.

Returns:

TCL_ERROR or TCL_OK

Example

```
iqtcl_QAMStartScan $connectID 0 30
```

See also:

5.13.1.7 iqtcl_IsQAMScanning

Check if QAM Cricket is scanning.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

Boolean true or false

Example

```
set bScanMode [ iqtcl_IsQAMScanning $connectId ]
```

See also:

5.13.1.8 iqtcl_QAMStartLearn

Starts the Learning phase for the QAM system.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_ERROR or TCL_OK

Example

```
iqtcl_QAMStartLearn $connectID
```

See also:

5.13.1.9 iqtcl_QAMStartOp

Start a QAM operation defined by passed in operation ID.

Note:

Operation IDs are defined as:

0	NO-OPERATION
3	LEARNING
4	TUNE
5	STREAM
6	SCAN
7	ALIAS

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***opID*** Integer defining the operation ID

← ***chanNum*** Integer defining the channel number to operate on, (or 0 for current active channel)

Returns:

Example

```
iqtcl_QAMStartOp $connectID 3 0
```

See also:

5.13.1.10 iqtcl_QAMTuneAndStreamBySTBChanName

Start a QAM tune and stream operation defined by passed in the STB channel name.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***char*** ∗ chanName A string representation of the STB Channel Name.

Returns:

Example

```
iqtcl_QAMTuneAndStreamBySTBChanName $connectID "Fox News"
```

See also:

5.13.1.11 iqtcl_QAMTuneAndStreamBySTBChanNumber

Start a QAM tune and stream operation defined by passed in the channel number.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***chan*** Integer defining the channel to tune and stream

Returns:

Example

```
iqtcl_QAMTuneAndStreamBySTBChanNumber $connectID 32
```

See also:

5.13.1.12 iqtcl_QAMStopStream

Stops the QAM system from streaming.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_ERROR or TCL_OK

Example

```
iqtcl_QAMStopStream $connectID
```

See also:

5.13.1.13 iqtcl_QAMStopScan

Stops the QAM from scanning.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_ERROR or TCL_OK

Example

```
iqtcl_QAMStopScan $connectID
```

See also:

5.13.1.14 iqtcl_QAMStopLearn

Stops the learning phase if it is currently ongoing.

Note:

The Stop Learn command is currently defined as a NO-OP. The Learning phase will be allowed to complete.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_ERROR or TCL_OK

Example

```
iqtcl_QAMStopLearn $connectID
```

See also:

5.13.1.15 iqtcl_QAMStopOp

Generically stops the current QAM operation.

Note:

If the QAM is in a Learning operation, the Learning phase will be allowed to run to completion.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_ERROR or TCL_OK

Example

```
iqtcl_QAMStopOp $connectID
```

See also:

5.14 Record Methods

Functions

- iqtcl_OpenRecord
- iqtcl_OpenStreamRecord
- iqtcl_CloseRecord
- iqtcl_StartRecord
- iqtcl_StopRecord
- iqtcl_UploadRecord
- iqtcl_SaveUploadRecordToFile
- iqtcl_GetRecordStatus
- iqtcl_SetTriggerPos
- iqtcl_AddTriggerCondition
- iqtcl_RemoveTriggerCondition
- iqtcl_GetTriggerStatus
- iqtcl_IsRecordTriggered
- iqtcl_StartRecord_TriggerPos

5.14.1 Detailed Description

Methods for Record

5.14.1.1 iqtcl_OpenRecord

Opens a new record session.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***size*** Size of record

Returns:

A 32-bit record handle or TCL_ERROR.

Example

```
set recordHandle [ iqtcl_OpenRecord $connectID $size ]
```

See also:

5.14.1.2 iqtcl_OpenStreamRecord

Opens a new record session setting a filter to capture only packets from a requested stream ID.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The stream ID to capture.

Returns:

A 32-bit record Handle or TCL_ERROR.

Example

```
 set recordHandle [ iqtclOpenStreamRecord $connectID $streamID ]
```

See also:

iqtcl_GetStreamID

5.14.1.3 iqtcl_CloseRecord

Closes the open record session.

Note:

Closing the record will release all resources reserved during the iqtcl_OpenRecord API method.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordID*** The record Handle ID returned from the iqtcl_OpenRecord API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
 iqtcl_CloseRecord $connectID $recordHandle
```

See also:

iqtcl_OpenRecord

5.14.1.4 iqtcl_StartRecord

Start recording events to a buffer on the target.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordHandle*** The record handle returned from the iqtcl_OpenRecord or iqtcl_OpenStreamRecord methods.

← ***flags*** a 32-bit flags bitmask value with the following bits defined:

1. 0x1: Circular/One Shot Mode
   (a) (Value = 1)One shot mode. Capture packets until buffer is full then stop.
   (b) (Value = 0)Capture into a circular buffer.
2. 0x2 = Record Headers
   (a) (Value = 1) Capture full event headers with data
   (b) (Value = 0) Capture event data bytes only

Returns:

Example

See also:

5.14.1.5 iqtcl_StopRecord

Stop recording event on the target.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordHandle*** The record Handle ID returned from the iqtcl_OpenRecord API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopRecord $connectID $recordHandle
```

See also:

5.14.1.6 iqtcl_UploadRecord

Upload the recorded event information from the target.

Note:

The record upload will be done via a TCP connection. The requesting application should open a socket and listen. This local socket port is passed to the target as part of this API call. The target will connect to this listening socket and start sending recorded event information. When the target has completed sending the data it will close the connection.

The recorded data arrives in the following format:

- 32 bit event ID: The event ID is a placeholder for future use. It can be ignored
- 64 bit timestamp: The timestamp has a 10ns resolution.
- 32 bit event data size: The data size is the total number of event data bytes to follow.
- n bits of event data: The event data may or may not include an event header depending on what flags were passed into the iqtcl_StartRecord API method.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordHandle*** The record Handle ID returned from the iqtcl_OpenRecord API method.

← ***tcport*** The socket port the target should connect to to start sending data.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_UploadRecord $connectID $recordHandle $tcport
```

See also:

5.14.1.7 iqtcl_SaveUploadRecordToFile

Note:

Parameters:

- ← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.
- ← ***recordHandle*** The record Handle ID returned from the iqtcl_OpenRecord API method.
- ← ***listenPort***
- ← ***recordSize***
- ← ***fileName***

Returns:

Example

```
iqtcl_SaveUploadRecordToFile $connectID $recordHandle $listenPort $recordSize $fileName
```

See also:

5.14.1.8 iqtcl_GetRecordStatus

Get the status of the record process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordID*** The record Handle ID returned from the iqtcl_OpenRecord API method.

Returns:

A Tcl_List structure or TCL_ERROR. The first item in the list is the Status flag defined as:
0 = INACTIVE - record has inactive and ready to upload data
1 = ACTIVE - record process is currently recording packets
2 = UPLOADING - record process is uploading data to host.
The second item in the list is the number of events in the record buffer.

Example

```
set statusList [ iqtcl_GetRecordStatus $connectID $recordHandle ]
```

See also:

iqtcl_OpenRecord

5.14.1.9 iqtcl_SetTriggerPos

iqtcl_SetTriggerPos mgb: is this DEPRECATED?? .. not supported by firmware, use iqtcl_StartRecord_TriggerPos instead

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

Example

See also:

5.14.1.10 iqtcl_AddTriggerCondition

Add a trigger condition to the record-on-trigger process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordID*** The record Handle ID returned from the iqtcl_OpenRecord API method.

← ***triggerType*** The trigger type to be added.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_AddTriggerCondition $connectId $recordId $triggerType
```

See also:

5.14.1.11 iqtcl_RemoveTriggerCondition

Remove a trigger condition to the record-on-trigger process.

Note:

Parameters:

- ← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.
- ← ***recordID*** The record Handle ID returned from the iqtcl_OpenRecord API method.
- ← ***triggerID*** The trigger handle to be removed.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_RemoveTriggerCondition $connectId $recordId $triggerType
```

See also:

5.14.1.12 iqtcl_GetTriggerStatus

Check the trigger status of for a recording.

Note:

Parameters:

← ***connectId*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordId*** The record Handle ID returned from the iqtcl_OpenRecord API method.

Returns:

A StringObj of the requested trigger status or TCL_ERROR. This string includes:

- Type
- Size
- Flags
- Pre-fill buffer capacity size in bytes
- Pre-fill buffer used size in bytes
- Post-fill buffer capacity size in bytes
- Post-fill buffer used size in bytes
- Handle

Example

```
set triggerStatus [ iqtcl_GetTriggerStatus $connectId $recordId ]
```

See also:

5.14.1.13 iqtcl_IsRecordTriggered

Check if record process has been triggered.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordID*** The record Handle ID returned from the iqtcl_OpenRecord API method.

Returns:

Boolean true or false

Example

```
set TriggeredStatus [ iqtcl_IsRecordTriggered $connectId $recordId ]
```

See also:

5.14.1.14 iqtcl_StartRecord_TriggerPos

Start a record process with a trigger position specification.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***recordID*** The record Handle ID returned from the iqtcl_OpenRecord API method.

← ***TriggerEnabled*** Trigger mode, values are:

- 0: one-shot mode, event data bytes only
- 1: circular mode

← ***preTriggerPerc*** Percentage of the buffer to use for pre-fill buffer

← ***postTriggerPerc*** Percentage of the buffer to use for post-fill buffer

Returns:

Example

To set a trigger buffer of 20% pre-fill, 80% post-fill:

```
set TrigEnabled   1
set preTrigPerc  20
set postTrigPerc 80
iqtcl_StartRecord_TriggerPos $connectID $recordID $TrigEnabled
                              $preTrigPerc $postTrigPerc
```

See also:

5.15 RVL Methods

Functions

- iqtcl_StartRVLScan
- iqtcl_StopRVLScan
- iqtcl_GetRVLStatus
- iqtcl_RVLLockStream
- iqtcl_RVLUnlockStream

5.15.1 Detailed Description

Methods for RVL

5.15.1.1 iqtcl_StartRVLScan

Start the RVL scanning process.

Note:

The RVL scanning process will pass out the locked in stream through the RVL port.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR

Example

```
 iqtcl_Start RVLScan
```

See also:

iqtcl_StopRVLScan
iqtcl_RVLLockStream

5.15.1.2 iqtcl_StopRVLScan

Stop the RVL scanning process.

Note:

The RVL scanning process will pass out the locked in stream through the RVL port.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

TCL_OK or TCL_ERROR.

Example

```
iqtcl_StopRVLScan
```

See also:

iqtcl_StartRVLScan

5.15.1.3 iqtcl_GetRVLStatus

Returns the status of the RVL process.

Note:

None.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A ByteArrayObj containing the data of the request or TCL_ERROR. The data array will contain the following structures:

- tTAPSTATUS

Example

```
set status [ iqtcl_GetRVLStatus $connectID ]
```

See also:

5.15.1.4 iqtcl_RVLLockStream

Locks the requested stream into the RVL process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The requested stream ID.

Returns:

TCL_OK or TCL_ERROR.

Example

```
iqtcl_RVLLockStream $connectID $streamID
```

See also:

iqtcl_RVLUnlockStream

5.15.1.5 iqtcl_RVLUnlockStream

Removes the requested stream from the RVL process.

Note:

Parameters:

- ← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.
- ← ***streamID*** The requested stream ID.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_RVLUnlockStream $connectID $streamID
```

See also:

iqtcl_RVLLockStream

5.16 Port Methods

Functions

- iqtcl_ClearPortCounters
- iqtcl_GetPortStatus
- iqtcl_GetPortCounters
- iqtcl_GetPortCounterByIndex
- iqtcl_GetPortCounterTableItem
- iqtcl_GetPortCounterTable

5.16.1 Detailed Description

Methods for PORT

5.16.1.1 iqtcl_ClearPortCounters

Clear the RMON stats from a particular port.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***portNum*** The port to clear. Valid values are:

- 0 = Port 1 (Main Port)
- 1 = Port 2 (TAP/RVL Port)

Returns:

TCL_OK or TCL_ERROR

Example

```
set port1 0
iqtcl_ClearPortCounters $connectID $port1
```

See also:

5.16.1.2 iqtcl_GetPortStatus

Retrieve the current status structure for a particular port.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***portNum*** The port counters to retrieve. Valid values are:

- 0 = Port 1 (Main Port)
- 1 = Port 2 (TAP/RVL Port)

Returns:

A ByteArrayObj containing the requested stats or TCL_ERROR. The data array will be the tPORTSTATE structure.

Example

```
set port1 0
set stats [ iqtcl_GetPortStatus $connectID $port1 ]
```

See also:

5.16.1.3 iqtcl_GetPortCounters

Retrieve the RMON stats from a particular port.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***portNum*** The port counters to retrieve. Valid values are:

- 0 = Port 1 (Main Port)
- 1 = Port 2 (TAP/RVL Port)

Returns:

A ByteArrayObj containing the requested stats or TCL_ERROR. The data array will be one of the following structures:

- tTRITENCTRS
- tNEMOCTRS
- tIXF18103CTRS

Example

```
set port1 0
set stats [ iqtcl_GetPortCounters $connectID $port1 ]
```

See also:

5.16.1.4 iqtcl_GetPortCounterByIndex

Retrieve a specific RMON counter from a particular port.

Note:

The index values are defined in the tcl file iq_defines.tcl. The available port counters may vary depending on the type of target you are connecting to.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***portNum*** The port to clear. Valid values are:

- 0 = Port 1 (Main Port)
- 1 = Port 2 (TAP/RVL Port)

← ***index*** The index of RMON counter desired

Returns:

A 64 bit value containing the requested stats value or TCL_ERROR.

Example

```
set port1 0
set index 0
set stats [ iqtcl_GetPortCounterByIndex $connectID $port1 $index ]
```

See also:

5.16.1.5 iqtcl_GetPortCounterTableItem

Retrieve a specific RMON counter from a particular port.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***portNum*** The port counters to retrieve. Valid values are:

- 0 = Port 1 (Main Port)
- 1 = Port 2 (TAP/RVL Port)

← ***index*** The index of RMON counter desired

← ***strSeps*** An optional separator string between name and value. If not defined the default separator value (TAB) will be used.

Returns:

A string containing a table of the requested stats or TCL_ERROR.

Example

```
set port1 0
set index 0
set strSep " : "
set strTable [ iqtcl_GetPortCounterTable $connectID $port1 $index $strSep ]
```

See also:

5.16.1.6 iqtcl_GetPortCounterTable

Retrieve a table of RMON stats from a particular port.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***portNum*** The port counters to retrieve. Valid values are:

- 0 = Port 1 (Main Port)
- 1 = Port 2 (TAP/RVL Port)

← ***strSep*** An optional separator string between name and value. If not defined the default separator value (TAB) will be used.

Returns:

A string containing a table of the requested stats or TCL_ERROR.

Example

```
set port1 0
set strSep " : "
set strTable [ iqtcl_GetPortCounterTable $connectID $port1 $strSep ]
```

See also:

5.17 Stimulus Methods

Functions

- iqtcl_GetStimulusStatus
- iqtcl_OpenStimulus
- iqtcl_OpenSmallStimulus
- iqtcl_CloseStimulus
- iqtcl_ReplicateStream
- iqtcl_DownloadStimulusFile
- iqtcl_LoadDefaultFile
- iqtcl_SetBackgroundTraffic
- iqtcl_DownloadLibpcapFile
- iqtcl_StartStimulus
- iqtcl_StopStimulus
- iqtcl_GetStimDiscovery
- iqtcl_StopStimDiscovery
- iqtcl_SetTracer
- iqtcl_ClearTracer

5.17.1 Detailed Description

Methods for Stimulus

5.17.1.1 iqtcl_GetStimulusStatus

Get the current status of the stimulus engine.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

A 32-bit value containing the current stimulus status. Current status values are defined as:

- 0 = Engine Stopped
- 1 = Engine Transmitting

Example

```
set stimStatus [ iqtcl_GetStimulusStatus $connectID $stimHandle ]
```

See also:

5.17.1.2 iqtcl_OpenStimulus

Open a new stimulus session.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A 32-bit stimulus handle or TCL_ERROR.

Example

```
set stimHandle [ iqtclOpenStimulus $connectID ]
```

See also:

5.17.1.3 iqtcl_OpenSmallStimulus

Open a stimulus session to one of the small capability engines.

Note:

The small capability engines have an internal buffer limit of 1000 bytes.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

Returns:

A 32-bit stimulus handle or TCL_ERROR.

Example

set stimHandle [iqtclOpenSmallStimulus $connectID]

See also:

5.17.1.4 iqtcl_CloseStimulus

Closes the open stimulus session.

Note:

Closing the stimulus session will release all resources reserved during the iqtcl_OpenStimulus API method.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
 iqtcl_CloseStimulus $connectID $stimHandle
```

See also:

iqtcl_OpenStimulus

5.17.1.5 iqtcl_ReplicateStream

Replicate an incoming live stream back out as stimulus.

Note:

Input values 4-13 are used to create a filter to capture only a specific stream to replicate.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***min_size*** The minimum number of bytes to buffer before starting stimulus.

← ***srcMAC*** The source MAC of the incoming stream, should use sent as a byte array.

← ***dstMAC*** The destination MAC of the incoming stream, should use sent as a byte array.

← ***vlanID*** The VLAN ID of the incoming stream should use

← ***TosField*** The TOS field of the incoming stream should use

← ***srcIP*** The source IP of the incoming stream should use sent as a string.

← ***dstIP*** The destination IP of the incoming stream should use sent as a string.

← ***srcPort*** The source port of the incoming stream should use

← ***dstPort*** The destination port of the incoming stream should use

← ***RtpType*** The RTP type of the incoming stream should use

← ***EncType*** The encapsulation type of the incoming stream should use. Valid values are:

- 0 = Eth2/IP/UDP encapsulation
- 1 = Eth2/IP/UDP/RTP encapsulation
- 2 = Eth2-VLAN/IP/UDP encapsulation
- 3 = Eth2-VLAN/IP/UDP/RTP encapsulation

← ***srcMAC*** The source MAC of the stimulus engine, should use sent as a byte array.

← ***dstMAC*** The destination MAC of the stimulus engine, should use sent as a byte array.

← ***vlanID*** The VLAN ID of the stimulus engine should use

← ***TosField*** The TOS field of the stimulus engine should use

← ***srcIP*** The source IP of the stimulus engine should use sent as a string.

← ***dstIP*** The destination IP of the stimulus engine should use sent as a string.

← ***srcPort*** The source port of the stimulus engine should use

← ***dstPort*** The destination port of the stimulus engine should use

← ***RtpType*** The RTP type of the stimulus engine should use

← ***EncType*** The encapsulation type of the stimulus engine (See 13)

Returns:

TCL_OK or TCL_ERROR

Example

```
See also:
```

5.17.1.6 iqtcl_DownloadStimulusFile

Download a data file for transmission.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***filepath*** The filename and path of data file

← ***srcMAC*** The source MAC the stimulus engine should use sent as a byte array.

← ***dstMAC*** The destination MAC the stimulus engine should use sent as a byte array.

← ***vlanID*** The VLAN ID the stimulus engine should use

← ***TosField*** The TOS field the stimulus engine should use

← ***srcIP*** The source IP the stimulus engine should use sent as a string.

← ***dstIP*** The destination IP the stimulus engine should use sent as a string.

← ***srcPort*** The source port the stimulus engine should use

← ***dstPort*** The destination port the stimulus engine should use

← ***RtpType*** The RTP type the stimulus engine should use

← ***EncType*** The encapsulation type the stimulus engine should use. Valid values are:

- 0 = Eth2/IP/UDP encapsulation
- 1 = Eth2/IP/UDP/RTP encapsulation
- 2 = Eth2-VLAN/IP/UDP encapsulation
- 3 = Eth2-VLAN/IP/UDP/RTP encapsulation

Any fields not used in the defined encapsulation type will be ignored.

Returns:

TCL_OK or TCL_ERROR

Example

```
set srcMac [ binary format H* 0008d0123456 ]
set dstMac [ binary format H* 0008d0133333 ]
iqtcl_DownloadStimulusFile $connectID $stimHandle /tmp/a.ts $srcMac $dstMac 0 0
                           "192.168.1.1" "192.168.2.2" 100 200 33 1
```

See also:

5.17.1.7 iqtcl_LoadDefaultFile

Load the default video file from the target's file system into the stimulus engine.

Note:

Part of the target's firmware installation is a MPEG-2 transport stream file for use as a default video file source.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***srcMac*** The source MAC the stimulus engine should use sent as a byte array.

← ***dstMac*** The destination MAC the stimulus engine should use sent as a byte array.

← ***vlanID*** The VLAN ID the stimulus engine should use

← ***TosField*** The TOS field the stimulus engine should use

← ***srcIP*** The source IP the stimulus engine should use sent as a string.

← ***dstIP*** The destination IP the stimulus engine should use sent as a string.

← ***srcPort*** The source port the stimulus engine should use

← ***dstPort*** The destination port the stimulus engine should use

← ***RtpType*** The RTP type the stimulus engine should use

← ***EncType*** The encapsulation type the stimulus engine should use. Valid values are:

- 0 = Eth2/IP/UDP encapsulation
- 1 = Eth2/IP/UDP/RTP encapsulation
- 2 = Eth2-VLAN/IP/UDP encapsulation
- 3 = Eth2-VLAN/IP/UDP/RTP encapsulation

Returns:

TCL_OK or TCL_ERROR

Example

```
set srcMac [ binary format H* 0008d0123456 ]
set dstMac [ binary format H* 0008d0133333 ]
iqtcl_LoadDefaultFile $connectID $stimHandle $srcMac $dstMac 0 0 "192.168.1.1" "192.168.2.
```

See also:

5.17.1.8 iqtcl_SetBackgroundTraffic

Download a background traffic format file for replay.

Note:

The background traffic file is created by the IQIPDataCreator application from IneoQuest Technologies Inc. There is no replication of a background traffic file.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***FilePath*** The filename and path of traffic format file.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_SetBackgroundTraffic $connectID $stimHandle /tmp/afile.iqd
```

See also:

5.17.1.9 iqtcl_DownloadLibpcapFile

Download a libpcap format file for replay.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***filename*** The filename and path of traffic format file.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_DownloadLibpcapFile $connectID $stimHandle /tmp/afile.pcap
```

See also:

5.17.1.10 iqtcl_StartStimulus

Start the stimulus engine.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***modmask*** The modmask is a bitmask representing which fields should be incremented if the stimulus stream is replicated. The following are the bitmask values:

- 0x1 = destination MAC
- 0x2 = source MAC
- 0x4 = VLAN ID
- 0x8 = source IP
- 0x10 = destination IP
- 0x20 = source Port
- 0x40 = destination Port
- 0x80 = RTP SSRCID field

← ***modvalue*** The modvalue is the increment value to apply to the fields defined by the modmask.

← ***portvalue*** The portvalue can be one of the following:

- 0 = Port 1 (Main Port)
- 1 = Port 2 (TAP/RVL Port)

← ***RecNum*** The number of records can be either the number of MTSPs per packet for a video stream or the number of bytes in a packet for voice streams.

← ***Stop*** The shouldloop can be one of the following:

- 0 = should loop
- 1 = should NOT loop

Returns:

TCL_OK or TCL_ERROR

Example

See also:

5.17.1.11 iqtcl_StopStimulus

Stop the stimulus engine from transmitting.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

TCL_OK or TCL_ERROR.

Example

```
iqtcl_StopStimulus $connectID $stimHandle
```

See also:

5.17.1.12 iqtcl_GetStimDiscovery

After downloading a data file, the target needs analyze the data file to determine stream information (data type, bitrate, etc). This call will start that process.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

A ByteArray Object identical to that returned from the iqtcl_GetCensusFirst API method.

Example

```
set censusData [ iqtcl_GetStimDiscovery $connectID $stimHandle ]
```

See also:

iqtcl_GetCensusFirst

5.17.1.13 iqtcl_StopStimDiscovery

Stop the target from continuing in the StimDiscovery phase for a downloaded file.

Note:

This method stops the process started by the iqtcl_GetStimDiscovery API call.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopStimDiscovery $connectID $stimHandle
```

See also:

5.17.1.14 iqtcl_SetTracer

Set the requested stream as a TRACER stream.

Note:

The stream ID is supplied as part of the census retrieval, or can be retrieved with the iqtcl_GetStreamID API call.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The stream ID to set.

Returns:

TCL_OK or TCL_ERROR

Example

iqtcl_SetTracer $connectID $streamID

See also:

iqtcl_GetStreamID

5.17.1.15 iqtcl_ClearTracer

Stop sending the stream as a TRACER stream.

Note:

The stream ID is supplied as part of the census retrieval, or can be retrieved with the iqtcl_GetStreamID API call.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***streamID*** The ID of stream to stop sending as tracer.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_ClearTracer $connectID $streamID
```

See also:

5.18 Stimulus Modification Methods

Functions

- iqtcl_SetXCount
- iqtcl_SetIPDrops
- iqtcl_SetIPJitter
- iqtcl_SetDFJitter
- iqtcl_SetPCRBitrate
- iqtcl_SetBitrate
- iqtcl_DropPid
- iqtcl_StopIPDrops
- iqtcl_StopIPJitter
- iqtcl_StopDFJitter
- iqtcl_StopDropPid

5.18.1 Detailed Description

Methods for Stimulus Modifications

5.18.1.1 iqtcl_SetXCount

Set the replication count for the stream.

Note:

Parameters:

- ← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.
- ← ***stimhandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.
- ← ***num*** The number of copies of the stream. Setting this value to 0 results in a single stream being transmitted.

Returns:

TCL_OK or TCL_ERROR

Example

To send 22 copies of a sream:

```
iqtcl_SetXCount $connectID $stimhandle 22
```

See also:

5.18.1.2 iqtcl_SetIPDrops

Drop IP Packets to cause packet loss to appear to the receiving stations.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***mode*** The drop mode to use - can be one of the following:

- 2 = Drop a specific number of packets (drop N)
- 4 = Drop a specific number of packets per packet count (drop N every X)

← ***drops*** The number of packets to drop (N)

← ***pass*** The number of packets to pass (X)

Returns:

TCL_OK or TCL_ERROR

Example

To drop 10 every 100 packets:

```
set mode   4
set drops 10
set pass  90
iqtcl_SetIPDrops $connectID $stimHandle $mode $drops $pass
```

See also:

5.18.1.3 iqtcl_SetIPJitter

Set the jitter value on the IP packets.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***mode*** The jitter mode - can be one of the following:

- 2 = Drop a specific number of packets (drop N)
- 4 = Drop a specific number of packets per packet count (drop N every X)

← ***drops*** The number of packets to drop (N)

← ***pass*** The number of packets to pass (X)

Returns:

TCL_OK or TCL_ERROR

Example

To drop 10 eve ry 100 packets:

```
set mode 3
set drops 10
set pass 100
iqtcl_SetIPDrops $connectID $stimHandle $mode $drops $pass
```

See also:

5.18.1.4 iqtcl_SetDFJitter

Set the current DF Jitter for the stream.

Note:

This will set a DF value for a stream. The stream will delay the arrival time of packets to achieve the desired DF value, and then burst other packets to maintain the expected bitrate.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***jitter*** The amount of DF jitter to add.

Returns:

TCL_OK or TCL_ERROR

Example

To set 100ms of DF Jitter:

```
iqtcl_SetDFJitter $connectID $stimHandle 100
```

See also:

5.18.1.5 iqtcl_SetPCRBitrate

Set the PCR bitrate of the stream.

Note:

This will vary the PCR bitrate away from the 27 Mhz default.

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimhandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***PcrBitrate*** The new PCR Bitrate in MHz.

Returns:

TCL_OK or TCL_ERROR

Example

To Set PCR bitrate to 28 MHz:

```
iqtcl_SetPCRBitrate $connectID $stimhandle 28
```

See also:

5.18.1.6 iqtcl_SetBitrate

Set the bitrate that the stream should be transmitted at.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimhandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***type*** The detection mode for the bitrare - can be one of the following values

- 0 = Default mode
- 1 = Detected Bitrate
- 2 = Received Bitrate
- 3 = User Defined Bitrate

← ***newBitrate*** The new bitrate of the stream

Returns:

TCL_OK or TCL_ERROR

Example

To change a video stream to 3.5 Mb/s:

```
set newBitrate 3500000
set type 3
iqtcl_SetBitrate $connectID $stimHandle $type $newBitrate
```

See also:

5.18.1.7 iqtcl_DropPid

Start dropping the designated Pid from the video stream.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

← ***pidID*** The Pid ID to drop.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_DropPid $connectID $stimHandle 0x81
```

See also:

5.18.1.8 iqtcl_StopIPDrops

Stop the IP Drops modification set using the iqtcl_SetIPDrops API method.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopIPDrops $connectID $stimHandle
```

See also:

5.18.1.9 iqtcl_StopIPJitter

Stop the IP Jitter modification set using the iqtcl_SetIPJitter API method.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopIPJitter $connectID $stimHandle
```

See also:

5.18.1.10 iqtcl_StopDFJitter

Stop the DF Jitter modification set using the iqtcl_SetDFJitter API method.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopDFJitter $connectID $stimHandle
```

See also:

5.18.1.11 iqtcl_StopDropPid

Stop the PID drop modification set using the iqtcl_DropPid API method.

Note:

Parameters:

← ***connectID*** The connection ID returned from the iqtcl_ConnectTo or iqtcl_OpenConnection methods.

← ***stimHandle*** The stimulus handle returned from the iqtcl_OpenStimulus API method.

Returns:

TCL_OK or TCL_ERROR

Example

```
iqtcl_StopDropPid $connectID $stimHandle
```

See also:

Chapter 6

Data Structure Documentation

6.1 tALARMHANDLE Struct Reference

6.1.1 Field Documentation

6.1.1.1 tTAG tag

6.1.1.2 char timestamp[32]

6.1.1.3 char handle[384]

6.1.1.4 char description[80]

6.2 tALARMINFO Struct Reference

6.2.1 Field Documentation

6.2.1.1 tTAG tag

6.2.1.2 UINT32 Id

6.2.1.3 UINT16 alarmId

6.2.1.4 UINT8 status

6.2.1.5 UINT8 severity

6.2.1.6 UINT32 streamId

6.2.1.7 UINT32 threshold

6.2.1.8 UINT32 value

6.2.1.9 WINT64 timestamp

6.3 tALIASCONFIG Struct Reference

Data Fields

- tTAG tag
- UINT32 id
- UINT32 srcIpAddress
- UINT32 destIpAddress
- UINT32 srcIpMask
- UINT32 destIpMask
- UINT16 srcPort
- UINT16 destPort
- char name [TEMPLATE_NAME_SIZE]
- UINT8 igmpStatus
- UINT8 modType
- UINT8 mac [6]
- char fmTemplate [TEMPLATE_NAME_SIZE]
- UINT16 fieldMask
- UINT8 bJoined
- UINT8 configStatus
- UINT32 ssrc
- UINT8 aliasType
- char charTemplate [TEMPLATE_NAME_SIZE]
- int vlanTci
- UINT8 videoType
- UINT8 tunerSdvType
- UINT32 tunerSdvMaxBw
- char tunerSdvDesc [TEMPLATE_NAME_SIZE]
- UINT32 intendedBitrate
- UINT8 intendedType
- UINT16 tsId
- UINT32 igmpSets
- UINT16 ports

6.3.1 Field Documentation

6.3.1.1 tTAG tag

structs TLV tag

6.3.1.2 UINT32 id

id

6.3.1.3 UINT32 srcIpAddress

source IP address

6.3.1.4 UINT32 destIpAddress

destination IP address

6.3.1.5 UINT32 srcIpMask

source netmask

6.3.1.6 UINT32 destIpMask

destination netmask

6.3.1.7 UINT16 srcPort

source port

6.3.1.8 UINT16 destPort

desination port

6.3.1.9 char name[TEMPLATE_NAME_SIZE]

alias name

6.3.1.10 UINT8 igmpStatus

IGMP status

6.3.1.11 UINT8 modType

modType

6.3.1.12 UINT8 mac[6]

MAC address

6.3.1.13 char fmTemplate[TEMPLATE_NAME_SIZE]

fm template

6.3.1.14 UINT16 fieldMask

field mask

6.3.1.15 UINT8 bJoined

boolean for joined IGMP flow

6.3.1.16 UINT8 configStatus

configStatus

6.3.1.17 UINT32 ssrc

ssrc

6.3.1.18 UINT8 aliasType

aliasType

6.3.1.19 char charTemplate[TEMPLATE_NAME_SIZE]

char template

6.3.1.20 int vlanTci

VLAN tag control info

6.3.1.21 UINT8 videoType

video type

6.3.1.22 UINT8 tunerSdvType

tunerSdvType

6.3.1.23 UINT32 tunerSdvMaxBw

tunerSdvMaxBw

6.3.1.24 char tunerSdvDesc[TEMPLATE_NAME_SIZE]

tunerSdvDesc

6.3.1.25 UINT32 intendedBitrate

intendedBitrate

6.3.1.26 UINT8 intendedType

intendedType

6.3.1.27 UINT16 tsId

transport stream ID

6.3.1.28 UINT32 igmpSets

IGMP set(s) of which this flow is a member

6.3.1.29 UINT16 ports

ports

6.4 tALIASNAME Struct Reference

6.4.1 Field Documentation

6.4.1.1 tTAG tag

6.4.1.2 char alias[80]

6.5 tASISTATUS Struct Reference

6.5.1 Field Documentation

6.5.1.1 tTAG tag

6.5.1.2 UINT32 status

6.5.1.3 UINT32 id

6.5.1.4 UINT32 hostIP

6.6 tCENTRY Struct Reference

6.6.1 Field Documentation

6.6.1.1 tTAG tag

6.6.1.2 UINT32 ID

6.6.1.3 WINT64 timestamp

6.6.1.4 UINT32 flags

6.6.1.5 UINT8 streamType

6.6.1.6 UINT8 hdrSize

6.6.1.7 UINT16 payloadSize

6.6.1.8 UINT32 bitrate

6.6.1.9 UINT32 detectedBitrate

6.6.1.10 UINT32 extFlags

6.7 tETHINFO Struct Reference

6.7.1 Field Documentation

6.7.1.1 tTAG tag

6.7.1.2 UINT8 destMac[6]

6.7.1.3 UINT8 srcMac[6]

6.8 tIGMPEVENT Struct Reference

6.8.1 Field Documentation

6.8.1.1 tTAG tag

6.8.1.2 UINT32 handle

6.8.1.3 UINT32 address

6.8.1.4 UINT32 vlan

6.8.1.5 UINT32 srcFilter

6.8.1.6 UINT32 minTime

6.8.1.7 UINT32 maxTime

6.8.1.8 UINT32 lastTime

6.8.1.9 UINT32 avgTime

6.8.1.10 UINT32 result[4]

6.8.1.11 struct { ... } join

6.8.1.12 struct { ... } leave

6.8.1.13 UINT8 state

6.9 tIGMPGROUPS Struct Reference

6.9.1 Field Documentation

6.9.1.1 tTAG tag

6.9.1.2 UINT32 address[2]

6.10 tIGMPSTATS Struct Reference

6.10.1 Field Documentation

6.10.1.1 tTAG tag

6.10.1.2 UINT32 handle

6.10.1.3 UINT32 flags

6.10.1.4 UINT32 address

6.10.1.5 UINT32 minTime

6.10.1.6 UINT32 maxTime

6.10.1.7 UINT32 lastTime

6.10.1.8 UINT32 avgTime

6.10.1.9 UINT32 nSuccess

6.10.1.10 UINT32 nFail

6.10.1.11 UINT32 vlan

6.10.1.12 UINT32 srcFilter

6.10.1.13 UINT32 lastLveTime

6.11 tIGMPSTATUS Struct Reference

6.11.1 Field Documentation

6.11.1.1 tTAG tag

6.11.1.2 UINT32 taskStatus

6.12 tIPINFO Struct Reference

6.12.1 Field Documentation

6.12.1.1 tTAG tag

6.12.1.2 UINT32 srcIP

6.12.1.3 UINT32 dstIP

6.12.1.4 UINT16 srcPort

6.12.1.5 UINT16 dstPort

6.12.1.6 UINT8 protocol

6.12.1.7 UINT8 tos

6.12.1.8 UINT16 vlanID

6.13 tIXF18103CTRS Struct Reference

Data Fields

- WINT64 TxTotalOctets
- WINT64 TxMulticastFrames
- WINT64 TxBroadcastFrames
- WINT64 TxTotalFrames
- WINT64 TxSizeFrames [7]
- WINT64 TxVLANFrames
- WINT64 TxPAUSECtrlFrames
- WINT64 TxUnicastFrames
- WINT64 TxMACCtrlFrames
- WINT64 RxTotalOctets
- WINT64 RxMulticastFrames
- WINT64 RxBroadcastFrames
- WINT64 RxTotalFrames
- WINT64 RxSizeFrames [7]
- WINT64 RxVLANFrames
- WINT64 RxPAUSECtrlFrames
- WINT64 RxUnicastFrames
- WINT64 RxMACCtrlFrames
- WINT64 RxEthUndersized
- WINT64 RxEthOversized
- WINT64 RxEthOctets
- WINT64 RxEthPkts
- WINT64 RxEthFragments
- WINT64 RxEthJabbers
- WINT64 RxEthFcs
- WINT64 TxTotalBytes
- WINT64 RxTotalBytes
- WINT64 RxBadFrames
- WINT64 RxGoodFrames

6.13.1 Field Documentation

6.13.1.1 tTAG tag

6.13.1.2 WINT64 TxTotalOctets

transmit_octets_counter - cleared on SNAP

6.13.1.3 WINT64 TxMulticastFrames

transmit_mcast_counter

6.13.1.4 WINT64 TxBroadcastFrames

transmit_bcast_counter

6.13.1.5 WINT64 TxTotalFrames

transmit_pkt_counter

6.13.1.6 WINT64 TxSizeFrames[7]

transmit_64_octets to transmit_MAX_octets_counter

6.13.1.7 WINT64 TxVLANFrames

transmit_VLAN_counter

6.13.1.8 WINT64 TxPAUSECtrlFrames

transmit_pausectl_counter

6.13.1.9 WINT64 TxUnicastFrames

transmit_unicast_counter

6.13.1.10 WINT64 TxMACCtrlFrames

transmit_MAC_ctl_counter

6.13.1.11 WINT64 RxTotalOctets

receive_octets_counter - cleared on SNAP

6.13.1.12 WINT64 RxMulticastFrames

receive_mcast_counter

6.13.1.13 WINT64 RxBroadcastFrames

receive_bcast_counter

6.13.1.14 WINT64 RxTotalFrames

receive_pkt_counter

6.13.1.15 WINT64 RxSizeFrames[7]

receive_64_octets to transmit_MAX_octets_counter

6.13.1.16 WINT64 RxVLANFrames

receive_VLAN_counter

6.13.1.17 WINT64 RxPAUSECtrlFrames

receive_pausectl_counter

6.13.1.18 WINT64 RxUnicastFrames

receive_unicast_counter

6.13.1.19 WINT64 RxMACCtrlFrames

receive_MAC_ctl_counter

6.13.1.20 WINT64 RxEthUndersized

receive_eth_undersize_pkt_counter

6.13.1.21 WINT64 RxEthOversized

receive_eth_oversize_pkt_counter

6.13.1.22 WINT64 RxEthOctets

receive_eth_octet_counter

6.13.1.23 WINT64 RxEthPkts

receive_eth_pkt_counter

6.13.1.24 WINT64 RxEthFragments

receive_eth_fragment_counter

6.13.1.25 WINT64 RxEthJabbers

receive_eth_jabber_counter

6.13.1.26 WINT64 RxEthFcs

receive_eth_fcs_error_counter

6.13.1.27 WINT64 TxTotalBytes

NOT cleared on snap

6.13.1.28 WINT64 RxTotalBytes

NOT cleared on snap

6.13.1.29 WINT64 RxBadFrames

RxBadFrames

6.13.1.30 WINT64 RxGoodFrames

RxGoodFrames

6.14 tMDIINFO Struct Reference

6.14.1 Field Documentation

6.14.1.1 tTAG tag

6.14.1.2 UINT32 nSamples

6.14.1.3 UINT32 dfMin

6.14.1.4 UINT32 dfMax

6.14.1.5 UINT32 dfCurrent

6.14.1.6 UINT32 dfAvg

6.14.1.7 WINT64 dfTotal

6.14.1.8 UINT32 mlMin

6.14.1.9 UINT32 mlMax

6.14.1.10 UINT32 mlCurrent

6.14.1.11 UINT32 mlAvg

6.14.1.12 WINT64 mlTotal

6.14.1.13 UINT32 ml15

6.14.1.14 UINT32 ml24

6.14.1.15 UINT32 vbMin

6.14.1.16 UINT32 vbMax

6.14.1.17 UINT32 vbCurrent

6.14.1.18 UINT32 vbAvg

6.14.1.19 WINT64 vbTotal

6.15 tMPEG2INFO Struct Reference

6.15.1 Field Documentation

6.15.1.1 tTAG tag

6.15.1.2 UINT16 networkPid

6.15.1.3 UINT8 nPrograms

6.15.1.4 UINT8 patVersion

6.15.1.5 UINT8 mtspSize

6.15.1.6 UINT32 nProgramAdded

6.15.1.7 UINT32 nProgramRemoved

6.15.1.8 UINT16 tsId

6.16 tMPEG2PID Struct Reference

6.16.1 Field Documentation

6.16.1.1 tTAG tag

6.16.1.2 UINT16 pid

6.16.1.3 UINT8 type

6.16.1.4 UINT8 flags

6.16.1.5 UINT32 nSamples

6.16.1.6 UINT32 pbrMin

6.16.1.7 UINT32 pbrMax

6.16.1.8 UINT32 pbrCurrent

6.16.1.9 UINT32 pbrAvg

6.16.1.10 WINT64 pbrTotal

6.16.1.11 UINT32 ccErrCurrent

6.16.1.12 UINT32 ccErrTotal

6.16.1.13 UINT32 extFlags

6.16.1.14 UINT32 outagePd

6.16.1.15 UINT32 lossRatio

6.16.1.16 UINT32 stateTime

6.16.1.17 UINT32 outages

6.16.1.18 UINT16 language

6.16.1.19 UINT16 misc

6.16.1.20 UINT8 duplicate[3]

6.17 tMPEG2PROGRAM Struct Reference

6.17.1 Field Documentation

6.17.1.1 tTAG tag

6.17.1.2 UINT16 nChannel

6.17.1.3 UINT8 nPids

6.17.1.4 char name[64]

6.17.1.5 char aliasName[TEMPLATE_NAME_SIZE]

6.17.1.6 UINT16 chanNumber

6.17.1.7 UINT8 progStatus

6.17.1.8 UINT8 alarmPids

6.17.1.9 char deviceRef[48]

6.17.1.10 UINT32 flags

6.17.1.11 UINT32 curBitrate

6.17.1.12 UINT32 stateTime

6.17.1.13 char providerName[64]

6.17.1.14 UINT32 totLoss

6.17.1.15 UINT16 curMlr

6.17.1.16 UINT16 firstPidIndex

6.17.1.17 UINT32 crc

6.18 tMTSPSTATS Struct Reference

6.18.1 Field Documentation

6.18.1.1 tTAG tag

6.18.1.2 UINT32 ccErrTotal

6.18.1.3 UINT32 ccErrCurrent

6.18.1.4 UINT32 syncError

6.18.1.5 UINT32 syncErrorTotal

6.18.1.6 UINT16 totalPids

6.18.1.7 UINT16 monPids

6.18.1.8 UINT16 almPids

6.18.1.9 UINT16 future

6.18.1.10 UINT32 nSamples

6.18.1.11 UINT32 pbrMin

6.18.1.12 UINT32 pbrMax

6.18.1.13 UINT32 pbrCurrent

6.18.1.14 UINT32 pbrAvg

6.18.1.15 WINT64 pbrTotal

6.19 tNAMETAG Struct Reference

6.19.1 Field Documentation

6.19.1.1 tTAG tag

6.19.1.2 char name[32]

6.20 tNemoCtrs Struct Reference

Data Fields

- WINT64 RxOctetsTotal
- WINT64 RxPacketsTotal
- WINT64 RxPacketsUnicast
- WINT64 RxPacketsMulticast
- WINT64 RxPacketsBroadcast
- WINT64 RxPackets [7]
- WINT64 TxOctetsTotal
- WINT64 TxPacketsTotal
- WINT64 TxPacketsUnicast
- WINT64 TxPacketsMulticast
- WINT64 TxPacketsBroadcast
- WINT64 TxPackets [7]
- WINT64 RxCrcError
- WINT64 RxErrors [3]

6.20.1 Field Documentation

6.20.1.1 WINT64 RxOctetsTotal

0x00

6.20.1.2 WINT64 RxPacketsTotal

0x08

6.20.1.3 WINT64 RxPacketsUnicast

0x10

6.20.1.4 WINT64 RxPacketsMulticast

0x18

6.20.1.5 WINT64 RxPacketsBroadcast

0x20

6.20.1.6 WINT64 RxPackets[7]

0x28

6.20.1.7 WINT64 TxOctetsTotal

0x60

6.20.1.8 WINT64 TxPacketsTotal

0x68

6.20.1.9 WINT64 TxPacketsUnicast

0x70

6.20.1.10 WINT64 TxPacketsMulticast

0x78

6.20.1.11 WINT64 TxPacketsBroadcast

0x80

6.20.1.12 WINT64 TxPackets[7]

0x88

6.20.1.13 WINT64 RxCrcError

0xC0

6.20.1.14 WINT64 RxErrors[3]

0xC8

6.21 tNEMOCTRS Struct Reference

6.21.1 Field Documentation

6.21.1.1 tTAG tag

6.21.1.2 struct tNemoCtrs ctrs `[read]`

6.22 tOldMPEG2PROGRAM Struct Reference

6.22.1 Field Documentation

6.22.1.1 tTAG tag

6.22.1.2 UINT16 nChannel

6.22.1.3 UINT16 pmtPid

6.22.1.4 UINT16 pcrPid

6.22.1.5 UINT8 nPids

6.23 tPMPIDSTATS Struct Reference

6.23.1 Field Documentation

6.23.1.1 tTAG tag

6.23.1.2 UINT32 maxBr

6.23.1.3 UINT32 minBr

6.23.1.4 UINT32 loss

6.23.1.5 UINT16 alarms

6.23.1.6 UINT16 ess

6.23.1.7 UINT32 ivlFaults

6.23.1.8 UINT32 avgBr

6.23.1.9 UINT16 outages

6.23.1.10 UINT16 outagePd

6.23.1.11 UINT32 ivlMask

6.23.1.12 UINT32 ivlHist

6.23.1.13 UINT32 ivlState

6.24 tPMPIDTOTALSTATS Struct Reference

6.24.1 Field Documentation

6.24.1.1 tTAG tag

6.24.1.2 UINT32 stopTime

6.24.1.3 UINT32 totalBr

6.24.1.4 UINT32 mlt24

6.24.1.5 UINT16 stateFlags

6.24.1.6 UINT16 totAlarms

6.24.1.7 UINT32 totEss

6.24.1.8 UINT32 totOutagePd

6.25 tPMPROGRAMIVLSTATS Struct Reference

Data Fields

- tTAG tag
- UINT32 maxBr
- UINT32 minBr
- UINT16 ess
- UINT8 outPids
- UINT8 alarms
- UINT16 maxMlr
- UINT16 mls15
- UINT32 mlt15
- UINT32 ivlMask
- UINT32 ivlFaults
- UINT32 ivlHist
- UINT8 progStatus
- UINT8 almPids
- UINT8 monPids
- UINT8 monOutPids
- UINT16 outages
- UINT16 outagePd
- UINT16 totOutagePd
- UINT16 totEss
- UINT16 totScteEvts
- UINT8 monitors
- UINT8 ivlFlags
- UINT32 avgBr
- UINT8 lastProgStatus
- UINT8 lossAlarms
- UINT16 totMls
- UINT16 maxMLp
- UINT16 minMLd
- UINT32 stateChanges

6.25.1 Field Documentation

6.25.1.1 tTAG tag

tag

6.25.1.2 UINT32 maxBr

maxBr

6.25.1.3 UINT32 minBr

minBr

6.25.1.4 UINT16 ess

ess

6.25.1.5 UINT8 outPids

outPids

6.25.1.6 UINT8 alarms

alarms

6.25.1.7 UINT16 maxMlr

maxMlr

6.25.1.8 UINT16 mls15

mls15

6.25.1.9 UINT32 mlt15

mlt15

6.25.1.10 UINT32 ivlMask

ivlMask

6.25.1.11 UINT32 ivlFaults

ivlFaults

6.25.1.12 UINT32 ivlHist

ivlHist

6.25.1.13 UINT8 progStatus

progStatus

6.25.1.14 UINT8 almPids

almPids

6.25.1.15 UINT8 monPids

monPids

6.25.1.16 UINT8 monOutPids

monPids

6.25.1.17 UINT16 outages

outagePd

6.25.1.18 UINT16 outagePd

outagePd

6.25.1.19 UINT16 totOutagePd

totEss

6.25.1.20 UINT16 totEss

totEss

6.25.1.21 UINT16 totScteEvts

totScteEvts

6.25.1.22 UINT8 monitors

monitors

6.25.1.23 UINT8 ivlFlags

ivlFlags

6.25.1.24 UINT32 avgBr

ivlFlags

6.25.1.25 UINT8 lastProgStatus

lastProgStatus

6.25.1.26 UINT8 lossAlarms

lastProgStatus

6.25.1.27 UINT16 totMls

totMls

6.25.1.28 UINT16 maxMLp

maxMLp

6.25.1.29 UINT16 minMLd

maxMLp

6.25.1.30 UINT32 stateChanges

maxMLp

6.26 tPMPROGRAMTOTSTATS Struct Reference

6.26.1 Field Documentation

6.26.1.1 tTAG tag

6.26.1.2 UINT32 mlt24

6.26.1.3 UINT32 ess

6.26.1.4 UINT32 mls24

6.26.1.5 UINT32 scteEvtTime

6.26.1.6 UINT32 stateCount

6.26.1.7 UINT32 pidStateCount

6.26.1.8 UINT32 totScteEvts

6.26.1.9 UINT32 maxMLp

6.26.1.10 UINT32 curMLp

6.26.1.11 UINT32 curMLd

6.26.1.12 UINT32 totEss

6.26.1.13 UINT32 totMLT

6.26.1.14 UINT32 totMLS

6.26.1.15 UINT32 totOutagePd

6.26.1.16 UINT32 totMaxMLp

6.26.1.17 UINT32 totMinMLd

6.27 tPMSTREAMGRAPHMETRICS Struct Reference

Data Fields

- tTAG tag
- UINT32 streamId
- UINT32 ivlTime
- UINT32 minBitRate
- UINT32 maxBitRate
- UINT32 pktLoss
- UINT32 mdiDf
- UINT16 rtpLd
- UINT16 Mls
- UINT16 Ess
- UINT16 Sess
- UINT16 Pess
- UINT16 starts
- UINT16 lastFlowPayldStatus
- UINT16 minPktRate
- UINT16 maxPktRate
- UINT16 minVBuffer
- UINT16 faultStatus
- UINT16 outagePd
- UINT32 faultTime
- UINT32 maxVBuffer
- UINT32 minVTsb
- UINT32 maxVTsb
- UINT16 lossProgCount
- UINT8 monPrograms
- UINT8 fltPrograms
- UINT8 monTsPids
- UINT8 fltTsPids
- UINT8 extFlags
- UINT8 flags
- UINT32 eMask [4]
- UINT32 eFaults [4]
- UINT32 eHistory [4]
- UINT32 rtpLoss
- UINT32 rtpLossEvts
- UINT16 rtplP
- UINT16 retryReqs
- UINT16 retryFills
- UINT16 usrFeebacks
- UINT16 rtpLs
- UINT16 flowPayldStatus
- float lossPercent
- UINT32 eStateChanges [4]

6.27.1 Field Documentation

6.27.1.1 tTAG tag

tag

6.27.1.2 UINT32 streamId

tag

6.27.1.3 UINT32 ivlTime

ivlTime

6.27.1.4 UINT32 minBitRate

minBitRate

6.27.1.5 UINT32 maxBitRate

maxBitRate

6.27.1.6 UINT32 pktLoss

pktLoss

6.27.1.7 UINT32 mdiDf

pktLoss

6.27.1.8 UINT16 rtpLd

rtpLd

6.27.1.9 UINT16 Mls

Mls

6.27.1.10 UINT16 Ess

Ess

6.27.1.11 UINT16 Sess

Sess

6.27.1.12 UINT16 Pess

Pess (formerly Uass)

6.27.1.13 UINT16 starts

starts

6.27.1.14 UINT16 lastFlowPayldStatus

lastFlowPayldStatus

6.27.1.15 UINT16 minPktRate

minPktRate

6.27.1.16 UINT16 maxPktRate

maxPktRate

6.27.1.17 UINT16 minVBuffer

maxPktRate

6.27.1.18 UINT16 faultStatus

faultStatus

6.27.1.19 UINT16 outagePd

outagePd

6.27.1.20 UINT32 faultTime

faultTime

6.27.1.21 UINT32 maxVBuffer

maxVBuffer

6.27.1.22 UINT32 minVTsb

minVTsb

6.27.1.23 UINT32 maxVTsb

lossProgCount

6.27.1.24 UINT16 lossProgCount

lossProgCount

6.27.1.25 UINT8 monPrograms

fltPrograms

6.27.1.26 UINT8 fltPrograms

fltPrograms

6.27.1.27 UINT8 monTsPids

monTsPids

6.27.1.28 UINT8 fltTsPids

fltTsPids

6.27.1.29 UINT8 extFlags

extFlags

6.27.1.30 UINT8 flags

flags

6.27.1.31 UINT32 eMask[4]

eMask

6.27.1.32 UINT32 eFaults[4]

eFaults

6.27.1.33 UINT32 eHistory[4]

eHistory

6.27.1.34 UINT32 rtpLoss

rtpLoss

6.27.1.35 UINT32 rtpLossEvts

rtpLossEvts

6.27.1.36 UINT16 rtplP

rtplP

6.27.1.37 UINT16 retryReqs

retryReqs

6.27.1.38 UINT16 retryFills

retryFills

6.27.1.39 UINT16 usrFeebacks

usrFeebacks

6.27.1.40 UINT16 rtpLs

rtpLs

6.27.1.41 UINT16 flowPayldStatus

flowPayldStatus

6.27.1.42 float lossPercent

lossPercent

6.27.1.43 UINT32 eStateChanges[4]

eStateChanges

6.28 tPMSTREAMMETRICS Struct Reference

6.28.1 Field Documentation

6.28.1.1 tTAG tag

6.28.1.2 UINT32 streamId

6.28.1.3 UINT32 Sdps

6.28.1.4 UINT32 Mls

6.28.1.5 UINT32 Ess

6.28.1.6 UINT32 Sess

6.28.1.7 UINT32 Uass

6.28.1.8 UINT32 starts

6.28.1.9 UINT32 faults

6.28.1.10 UINT32 flags

6.29 tPMSTREAMTOTALMETRICS Struct Reference

Data Fields

- tTAG tag
- UINT32 streamId
- UINT16 progNoAliasCnt
- UINT16 progAliases
- UINT32 Mls24
- UINT32 Ess
- UINT32 Sess
- UINT32 Pess
- UINT32 Actss
- UINT32 Totss
- UINT32 Totsts
- UINT32 pktLoss
- UINT32 outagePd
- UINT32 outageCt
- UINT32 Mls
- UINT32 usrQos
- UINT32 ls24
- UINT32 rtpLoss24
- float lossPercent
- UINT32 stateTime
- UINT32 progStateCount
- UINT32 flowStateCount
- UINT32 ledToFaultMap
- UINT32 mgtId
- UINT32 totOutagePd
- UINT32 totMloss
- UINT32 totEss
- UINT32 totPess

6.29.1 Field Documentation

6.29.1.1 tTAG tag

tag

6.29.1.2 UINT32 streamId

streanId

6.29.1.3 UINT16 progNoAliasCnt

progNoAliasCnt

6.29.1.4 UINT16 progAliases

progAliases

6.29.1.5 UINT32 Mls24

Mls24

6.29.1.6 UINT32 Ess

Ess

6.29.1.7 UINT32 Sess

Sess

6.29.1.8 UINT32 Pess

Pess (formerly Uass)

6.29.1.9 UINT32 Actss

Actss

6.29.1.10 UINT32 Totss

Totss

6.29.1.11 UINT32 Totsts

Totsts

6.29.1.12 UINT32 pktLoss

outagePd

6.29.1.13 UINT32 outagePd

outagePd

6.29.1.14 UINT32 outageCt

outageCt

6.29.1.15 UINT32 Mls

Mls

6.29.1.16 UINT32 usrQos

usrQos

6.29.1.17 UINT32 ls24

ls24

6.29.1.18 UINT32 rtpLoss24

rtpLoss24

6.29.1.19 float lossPercent

lossPercent

6.29.1.20 UINT32 stateTime

stateTime

6.29.1.21 UINT32 progStateCount

progStateCount

6.29.1.22 UINT32 flowStateCount

flowStateCount

6.29.1.23 UINT32 ledToFaultMap

ledToFaultMap

6.29.1.24 UINT32 mgtId

mgtId

6.29.1.25 UINT32 totOutagePd

total Stats

6.29.1.26 UINT32 totMloss

totMloss

6.29.1.27 UINT32 totEss

totEss

6.29.1.28 UINT32 totPess

totPess

6.30 tPMSYSTEMMETRICS Struct Reference

Data Fields

- tTAG tag
- UINT32 ivlTime
- UINT16 tNewStreams [12]
- UINT16 tBadStreams [4]
- UINT16 tMaxStreams [4]
- UINT16 tMinStreams [4]
- UINT32 mediaLoss
- UINT32 fltMapChanged
- UINT16 blueStreams [MAX_SYS_VIDEO_TYPES]
- UINT16 greyStreams [MAX_SYS_VIDEO_TYPES]
- UINT16 greenStreams [MAX_SYS_VIDEO_TYPES]
- UINT16 redStreams [MAX_SYS_VIDEO_TYPES]
- UINT16 orangeStreams [MAX_SYS_VIDEO_TYPES]
- UINT16 util [MAX_SYS_VIDEO_TYPES]
- UINT16 activeStreams [MAX_SYS_VIDEO_TYPES]
- UINT16 usrQos
- UINT16 retryReqs
- UINT16 evtsP0 [3]
- UINT16 evtsP1 [3]
- UINT16 retryFills
- UINT16 mls
- UINT16 maxLp
- UINT16 minLp
- UINT16 lpErrors
- UINT16 minLd
- UINT32 ipLoss
- UINT16 bcastStreams
- UINT16 evtsP2 [3]
- UINT8 systemStatus
- UINT8 flags
- UINT16 trapSentRate
- UINT32 timestamp

6.30.1 Field Documentation

6.30.1.1 tTAG tag

tag

6.30.1.2 UINT32 ivlTime

ivlTime

6.30.1.3 UINT16 tNewStreams[12]

tNewStreams

6.30.1.4 UINT16 tBadStreams[4]

tBadStreams

6.30.1.5 UINT16 tMaxStreams[4]

tMaxStreams

6.30.1.6 UINT16 tMinStreams[4]

tMinStreams

6.30.1.7 UINT32 mediaLoss

mediaLoss

6.30.1.8 UINT32 fltMapChanged

fltMapChanged

6.30.1.9 UINT16 blueStreams[MAX_SYS_VIDEO_TYPES]

blueStreams

6.30.1.10 UINT16 greyStreams[MAX_SYS_VIDEO_TYPES]

greyStreams

6.30.1.11 UINT16 greenStreams[MAX_SYS_VIDEO_TYPES]

greenStreams

6.30.1.12 UINT16 redStreams[MAX_SYS_VIDEO_TYPES]

redStreams

6.30.1.13 UINT16 orangeStreams[MAX_SYS_VIDEO_TYPES]

orangeStreams

6.30.1.14 UINT16 util[MAX_SYS_VIDEO_TYPES]

util

6.30.1.15 UINT16 activeStreams[MAX_SYS_VIDEO_TYPES]

activeStreams

6.30.1.16 UINT16 usrQos

usrQos

6.30.1.17 UINT16 retryReqs

retryReqs

6.30.1.18 UINT16 evtsP0[3]

evtsP0

6.30.1.19 UINT16 evtsP1[3]

evtsP1

6.30.1.20 UINT16 retryFills

retryFills

6.30.1.21 UINT16 mls

mls

6.30.1.22 UINT16 maxLp

maxLp

6.30.1.23 UINT16 minLp

minLp

6.30.1.24 UINT16 lpErrors

lpErrors

6.30.1.25 UINT16 minLd

minLd

6.30.1.26 UINT32 ipLoss

ipLoss

6.30.1.27 UINT16 bcastStreams

bcastStreams

6.30.1.28 UINT16 evtsP2[3]

evtsP2

6.30.1.29 UINT8 systemStatus

systemStatus

6.30.1.30 UINT8 flags

flags

6.30.1.31 UINT16 trapSentRate

trapSentRate

6.30.1.32 UINT32 timestamp

timestamp

6.31 tRTPINFO Struct Reference

6.31.1 Field Documentation

6.31.1.1 tTAG tag

6.31.1.2 UINT8 payloadType

6.32 tRTPSTATS Struct Reference

6.32.1 Field Documentation

6.32.1.1 tTAG tag

6.32.1.2 UINT32 seqErrTotal

6.32.1.3 UINT32 seqErrCurrent

6.32.1.4 UINT32 ldMin

6.32.1.5 UINT32 lpMax

6.32.1.6 UINT32 ldCurrent

6.32.1.7 UINT32 lpCurrent

6.32.1.8 UINT32 ldErrors

6.32.1.9 UINT32 lpErrors

6.32.1.10 UINT32 lossDuration

6.32.1.11 UINT32 lossEvtCurrent

6.32.1.12 UINT32 lossEvtTotal

6.32.1.13 float lossPercent

6.32.1.14 UINT32 dupCurrent

6.32.1.15 UINT32 dupTotal

6.32.1.16 UINT32 oosCurrent

6.32.1.17 UINT32 oosTotal

6.33 tSTREAMSTATS Struct Reference

Data Fields

- tTAG tag
- UINT32 nSamples
- UINT16 pktSizeMin
- UINT16 pktSizeMax
- UINT32 lbrMin
- UINT32 lbrMax
- UINT32 lbrCurrent
- UINT32 lbrAvg
- WINT64 lbrTotal
- UINT32 mbrMin
- UINT32 mbrMax
- UINT32 mbrCurrent
- UINT32 mbrAvg
- WINT64 mbrTotal
- UINT32 utilMin
- UINT32 utilMax
- UINT32 utilCurrent
- UINT32 utilAvg
- WINT64 utilTotal
- UINT32 faultStatus
- UINT32 faultMap
- UINT32 faultHistory
- UINT32 faultTime
- UINT32 decayCount
- UINT8 tos
- UINT32 userFeedback
- UINT32 pktMin
- UINT32 pktMax
- UINT32 pktCurrent
- UINT32 pktAvg
- WINT64 pktTotal

6.33.1 Field Documentation

6.33.1.1 tTAG tag

tag

6.33.1.2 UINT32 nSamples

nSamples

6.33.1.3 UINT16 pktSizeMin

pktSizeMin

6.33.1.4 UINT16 pktSizeMax

pktSizeMax

6.33.1.5 UINT32 lbrMin

line byte rate

6.33.1.6 UINT32 lbrMax

lbrMax

6.33.1.7 UINT32 lbrCurrent

lbrCurrent

6.33.1.8 UINT32 lbrAvg

lbrAvg

6.33.1.9 WINT64 lbrTotal

lbrTotal

6.33.1.10 UINT32 mbrMin

media byte rate

6.33.1.11 UINT32 mbrMax

mbrMax

6.33.1.12 UINT32 mbrCurrent

mbrCurrent

6.33.1.13 UINT32 mbrAvg

mbrAvg

6.33.1.14 WINT64 mbrTotal

mbrTotal

6.33.1.15 UINT32 utilMin

stream utilization

6.33.1.16 UINT32 utilMax

utilMax

6.33.1.17 UINT32 utilCurrent

utilCurrent

6.33.1.18 UINT32 utilAvg

utilAvg

6.33.1.19 WINT64 utilTotal

Stream Fault Status

6.33.1.20 UINT32 faultStatus

Deprecated. Alarms are NOW 64 bits

6.33.1.21 UINT32 faultMap

we can use this space(16 bytes) for any future.

6.33.1.22 UINT32 faultHistory

stream level data

6.33.1.23 UINT32 faultTime

faultTime

6.33.1.24 UINT32 decayCount

decayCount

6.33.1.25 UINT8 tos

latest TOS field

6.33.1.26 UINT32 userFeedback

button presses

6.33.1.27 UINT32 pktMin

packet count

6.33.1.28 UINT32 pktMax

pktMax

6.33.1.29 UINT32 pktCurrent

pktCurrent

6.33.1.30 UINT32 pktAvg

pktAvg

6.33.1.31 WINT64 pktTotal

pktTotal

6.34 tTAG Struct Reference

6.34.1 Field Documentation

6.34.1.1 UINT16 type

6.34.1.2 UINT16 size

6.35 tTAPSTATUS Struct Reference

6.35.1 Field Documentation

6.35.1.1 tTAG tag

6.35.1.2 UINT32 status

6.35.1.3 UINT32 id

6.35.1.4 UINT32 hostIP

6.36 tTARGETINFO Struct Reference

6.36.1 Field Documentation

6.36.1.1 tTAG tag

6.36.1.2 UINT32 targetMode

6.36.1.3 WINT64 timeDate

6.36.1.4 char targetName[32]

6.36.1.5 char targetLocation[32]

6.36.1.6 char targetContact[32]

6.37 tTRITENCTRS Struct Reference

6.37.1 Field Documentation

6.37.1.1 tTAG tag

6.37.1.2 tTritenCtrs ctrs

6.38 tTritenCtrs Struct Reference

Data Fields

- WINT64 RxOctetsGood
- WINT64 RxOctetsBad
- WINT64 RxPacketsUnicast
- WINT64 RxPacketsMulticast
- WINT64 RxPacketsBroadcast
- WINT64 RxPackets [7]
- WINT64 RxFcsErrors
- WINT64 RxTagged
- WINT64 RxDataErrors
- WINT64 RxAlignErrors
- WINT64 RxLongErrors
- WINT64 RxJabberErrors
- WINT64 RxPauseControl
- WINT64 RxUnknownControl
- WINT64 RxVeryLongErrors
- WINT64 RxRuntErrors
- WINT64 RxShortErrors
- WINT64 CarrierExtendErrors
- WINT64 RxSequenceErrors
- WINT64 RxSymbolErrors
- WINT64 RxTotalOctets
- WINT64 RxTotalPackets
- WINT64 RxTotalErrors
- WINT64 TxOctetsGood
- WINT64 TxOctetsBad
- WINT64 TxPacketsUnicast
- WINT64 TxPacketsMulticast
- WINT64 TxPacketsBroadcast
- WINT64 TxPackets [7]
- WINT64 TxDeffered
- WINT64 TxTotalCollisions
- WINT64 TxSingleCollisions
- WINT64 TxMultipleCollisions
- WINT64 TxLateCollisions
- WINT64 TxExcessiveCollisionErrors
- WINT64 TxExcessiveDefferalErrors
- WINT64 TxExcessiveLengthDrop
- WINT64 TxUnderrun
- WINT64 TxTagged
- WINT64 TxFcsErrors
- WINT64 TxPauseFrames
- WINT64 TxFlowControlCollisions
- WINT64 TxTotalOctets
- WINT64 TxTotalPackets
- WINT64 TxTotalErrors
- WINT64 TapPacketsDropped
- WINT64 TapBytesDropped

6.38.1 Field Documentation

6.38.1.1 WINT64 RxOctetsGood

RxOctetsGood

6.38.1.2 WINT64 RxOctetsBad

RxOctetsGood

6.38.1.3 WINT64 RxPacketsUnicast

RxPacketsUnicast

6.38.1.4 WINT64 RxPacketsMulticast

RxPacketsUnicast

6.38.1.5 WINT64 RxPacketsBroadcast

RxPacketsUnicast

6.38.1.6 WINT64 RxPackets[7]

RxPackets

6.38.1.7 WINT64 RxFcsErrors

RxPackets

6.38.1.8 WINT64 RxTagged

RxTagged

6.38.1.9 WINT64 RxDataErrors

RxTagged

6.38.1.10 WINT64 RxAlignErrors

RxAlignErrors

6.38.1.11 WINT64 RxLongErrors

RxLongErrors

6.38.1.12 WINT64 RxJabberErrors

RxJabberErrors

6.38.1.13 WINT64 RxPauseControl

RxPauseControl

6.38.1.14 WINT64 RxUnknownControl

RxUnknownControl

6.38.1.15 WINT64 RxVeryLongErrors

RxVeryLongErrors

6.38.1.16 WINT64 RxRuntErrors

RxRuntErrors

6.38.1.17 WINT64 RxShortErrors

RxShortErrors

6.38.1.18 WINT64 CarrierExtendErrors

CarrierExtendErrors

6.38.1.19 WINT64 RxSequenceErrors

RxSequenceErrors

6.38.1.20 WINT64 RxSymbolErrors

RxSymbolErrors

6.38.1.21 WINT64 RxTotalOctets

RxTotalOctets

6.38.1.22 WINT64 RxTotalPackets

RxTotalPackets

6.38.1.23 WINT64 RxTotalErrors

RxTotalErrors

6.38.1.24 WINT64 TxOctetsGood

TxOctetsGood

6.38.1.25 WINT64 TxOctetsBad

TxOctetsBad

6.38.1.26 WINT64 TxPacketsUnicast

TxPacketsUnicast

6.38.1.27 WINT64 TxPacketsMulticast

TxPacketsMulticast

6.38.1.28 WINT64 TxPacketsBroadcast

TxPacketsBroadcast

6.38.1.29 WINT64 TxPackets[7]

TxPackets

6.38.1.30 WINT64 TxDeffered

TxDeffered

6.38.1.31 WINT64 TxTotalCollisions

TxTotalCollisions

6.38.1.32 WINT64 TxSingleCollisions

TxSingleCollisions

6.38.1.33 WINT64 TxMultipleCollisions

TxMultipleCollisions

6.38.1.34 WINT64 TxLateCollisions

TxLateCollisions

6.38.1.35 WINT64 TxExcessiveCollisionErrors

TxExcessiveCollisionErrors

6.38.1.36 WINT64 TxExcessiveDefferalErrors

TxExcessiveDefferalErrors

6.38.1.37 WINT64 TxExcessiveLengthDrop

TxExcessiveLengthDrop

6.38.1.38 WINT64 TxUnderrun

TxUnderrun

6.38.1.39 WINT64 TxTagged

TxTagged

6.38.1.40 WINT64 TxFcsErrors

TxFcsErrors

6.38.1.41 WINT64 TxPauseFrames

TxPauseFrames

6.38.1.42 WINT64 TxFlowControlCollisions

TxFlowControlCollisions

6.38.1.43 WINT64 TxTotalOctets

TxTotalOctets

6.38.1.44 WINT64 TxTotalPackets

TxTotalPackets

6.38.1.45 WINT64 TxTotalErrors

TxTotalErrors

6.38.1.46 WINT64 TapPacketsDropped

TapPacketsDropped

6.38.1.47 WINT64 TapBytesDropped

TapBytesDropped

6.39 tTUNERRFMAP Struct Reference

6.39.1 Field Documentation

6.39.1.1 tTAG tag

6.39.1.2 UINT32 num

6.39.1.3 UINT32 freq

6.39.1.4 UINT32 symbolRate

6.40 tTUNERSTATS Struct Reference

Data Fields

- UINT32 freq
- UINT8 chan
- UINT8 signal
- UINT8 mod
- UINT8 id
- UINT8 active
- UINT32 nSamples
- UINT32 snrMin
- UINT32 snrMax
- UINT32 snrCurrent
- UINT32 snrAvg
- WINT64 snrTotal
- UINT32 rsUcCurrent
- UINT32 rsUcTotal
- UINT32 rsCoCurrent
- UINT32 rsCoTotal
- char chanPre
- UINT32 rsUcPercent
- UINT32 rsCoPercent
- WINT64 rsBytesTotal
- UINT8 tuner
- UINT8 align
- UINT32 flags
- UINT32 nSamplesRxPwr
- INT32 rxPwrMin
- INT32 rxPwrMax
- INT32 rxPwrCurrent
- INT32 rxPwrAvg
- INT64 rxPwrTotal
- UINT32 nSamplesBer
- float berPreMin
- float berPreMax
- float berPreCurrent
- float berPreAvg
- float berPreTotal
- float berPostMin
- float berPostMax
- float berPostCurrent
- float berPostAvg
- float berPostTotal

6.40.1 Field Documentation

6.40.1.1 tTAG tag

6.40.1.2 UINT32 freq

frequency (Hz)

6.40.1.3 UINT8 chan

channel

6.40.1.4 UINT8 signal

signal status

6.40.1.5 UINT8 mod

modulation

6.40.1.6 UINT8 id

identifier

6.40.1.7 UINT8 active

active/tuned flag

6.40.1.8 UINT32 nSamples

nSamples)

6.40.1.9 UINT32 snrMin

SNR min (dB x 1000)

6.40.1.10 UINT32 snrMax

SNR max (dB x 1000)

6.40.1.11 UINT32 snrCurrent

SNR current (dB x 1000)

6.40.1.12 UINT32 snrAvg

SNR average (dB x 1000)

6.40.1.13 WINT64 snrTotal

SNR total

6.40.1.14 UINT32 rsUcCurrent

Reed-Solomon uncorrected errors current

6.40.1.15 UINT32 rsUcTotal

Reed-Solomon uncorrected errors total

6.40.1.16 UINT32 rsCoCurrent

Reed-Solomon corrected errors current

6.40.1.17 UINT32 rsCoTotal

Reed-Solomon corrected errors total

6.40.1.18 char chanPre

optional channel number prefix character

6.40.1.19 UINT32 rsUcPercent

Reed-Solomon uncorrected percent x 10000

6.40.1.20 UINT32 rsCoPercent

Reed-Solomon corrected percent x 10000

6.40.1.21 WINT64 rsBytesTotal

rsBytesTotal

6.40.1.22 UINT8 tuner

tuner number if more than one tuner/port

6.40.1.23 UINT8 align

align

6.40.1.24 UINT32 flags

flags

6.40.1.25 UINT32 nSamplesRxPwr

nSamplesRxPwr

6.40.1.26 INT32 rxPwrMin

Rx power min (dBmV x 10)

6.40.1.27 INT32 rxPwrMax

Rx power max (dBmV x 10)

6.40.1.28 INT32 rxPwrCurrent

Rx power current (dBmV x 10)

6.40.1.29 INT32 rxPwrAvg

Rx power average (dBmV x 10)

6.40.1.30 INT64 rxPwrTotal

Rx power total (dBmV x 10)

6.40.1.31 UINT32 nSamplesBer

nSamplesBer

6.40.1.32 float berPreMin

BER pre-FEC min

6.40.1.33 float berPreMax

BER pre-FEC max

6.40.1.34 float berPreCurrent

BER pre-FEC current

6.40.1.35 float berPreAvg

BER pre-FEC average

6.40.1.36 float berPreTotal

BER pre-FEC total

6.40.1.37 float berPostMin

BER post-FEC min

6.40.1.38 float berPostMax

BER post-FEC max

6.40.1.39 float berPostCurrent

BER post-FEC current

6.40.1.40 float berPostAvg

BER post-FEC average

6.40.1.41 float berPostTotal

BER post-FEC total

Index

active
- tTUNERSTATS, 264

activeStreams
- tPMSYSTEMMETRICS, 243

Actss
- tPMSTREAMTOTALMETRICS, 239

address
- tIGMPEVENT, 206
- tIGMPGROUPS, 207
- tIGMPSTATS, 208

ALARM
- iqtcl_ClearAlarmLog, 24
- iqtcl_ClearAlarms, 23
- iqtcl_GetActiveAlarmFirst, 25
- iqtcl_GetActiveAlarmNext, 26
- iqtcl_GetActiveAlarmTable, 27
- iqtcl_GetAlarmLogFirst, 28
- iqtcl_GetAlarmLogNext, 29
- iqtcl_GetAlarmLogTable, 30

Alarm Methods, 22

alarmId
- tALARMINFO, 197

alarmPids
- tMPEG2PROGRAM, 218

alarms
- tPMPIDSTATS, 225
- tPMPROGRAMIVLSTATS, 228

ALIAS
- iqtcl_ClearAliases, 34
- iqtcl_GetAliasFirst, 32
- iqtcl_GetAliasNext, 33

alias
- tALIASNAME, 202

Alias Methods, 31

aliasName
- tMPEG2PROGRAM, 218

aliasType
- tALIASCONFIG, 200

align
- tTUNERSTATS, 265

almPids
- tMTSPSTATS, 219
- tPMPROGRAMIVLSTATS, 228

ARP Proxy Methods, 91

ARPPROXY
- iqtcl_AddArpProxy, 92
- iqtcl_AddArpProxyEx, 95
- iqtcl_ClearArpProxyTable, 94
- iqtcl_RemoveArpProxy, 93
- iqtcl_RemoveArpProxyEx, 96

ASI
- iqtcl_ASILockStream, 39
- iqtcl_ASIUnlockStream, 40
- iqtcl_GetASIStatus, 38
- iqtcl_StartASIScan, 36
- iqtcl_StopASIScan, 37

ASI Methods, 35

avgBr
- tPMPIDSTATS, 225
- tPMPROGRAMIVLSTATS, 229

avgTime
- tIGMPEVENT, 206
- tIGMPSTATS, 208

bcastStreams
- tPMSYSTEMMETRICS, 245

berPostAvg
- tTUNERSTATS, 267

berPostCurrent
- tTUNERSTATS, 267

berPostMax
- tTUNERSTATS, 267

berPostMin
- tTUNERSTATS, 267

berPostTotal
- tTUNERSTATS, 267

berPreAvg
- tTUNERSTATS, 266

berPreCurrent
- tTUNERSTATS, 266

berPreMax
- tTUNERSTATS, 266

berPreMin
- tTUNERSTATS, 266

berPreTotal
- tTUNERSTATS, 266

bitrate
- tCENTRY, 204

bJoined
- tALIASCONFIG, 199

blueStreams
    tPMSYSTEMMETRICS, 243

CAPTURE
    iqtcl_CloseCapture, 44
    iqtcl_GetCaptureStatus, 48
    iqtcl_OpenCapture, 42
    iqtcl_OpenStreamCapture, 43
    iqtcl_StartCapture, 45
    iqtcl_StopCapture, 46
    iqtcl_UploadCapture, 47
Capture Methods, 41
CarrierExtendErrors
    tTritenCtrs, 258
ccErrCurrent
    tMPEG2PID, 217
    tMTSPSTATS, 219
ccErrTotal
    tMPEG2PID, 217
    tMTSPSTATS, 219
CENSUS
    iqtcl_ClearCensus, 55
    iqtcl_ClearStream, 56
    iqtcl_ClearStreamStats, 57
    iqtcl_GetCensusByID, 52
    iqtcl_GetCensusFirst, 50
    iqtcl_GetCensusNext, 51
    iqtcl_GetCensusTable, 63
    iqtcl_GetCensusTableItem, 58
    iqtcl_GetCensusTableItemTagged, 59
    iqtcl_GetNextCensusID, 53
    iqtcl_GetPidTableTagged, 65
    iqtcl_GetProgramTableTagged, 64
    iqtcl_GetQAMTable, 62
    iqtcl_GetQAMTableItem, 60
    iqtcl_GetQAMTableItemTagged, 61
    iqtcl_GetStreamID, 54
Census Methods, 49
chan
    tTUNERSTATS, 263
chanNumber
    tMPEG2PROGRAM, 218
chanPre
    tTUNERSTATS, 265
charTemplate
    tALIASCONFIG, 200
configStatus
    tALIASCONFIG, 200
CONNECTION
    iqtcl_CloseConnection, 70
    iqtcl_CloseOldConnections, 74
    iqtcl_CloseSession, 72
    iqtcl_ConnectTo, 67
    iqtcl_IsConnected, 71
    iqtcl_OpenConnection, 68
    iqtcl_OpenConnectionWithMsgSize, 69
    iqtcl_ResumeSession, 73
    iqtcl_RetrieveSessionData, 76
    iqtcl_SaveSessionData, 75
Connection Methods, 66
crc
    tMPEG2PROGRAM, 218
ctrs
    tNEMOCTRS, 223
    tTRITENCTRS, 255
curBitrate
    tMPEG2PROGRAM, 218
curMLd
    tPMPROGRAMTOTSTATS, 231
curMLp
    tPMPROGRAMTOTSTATS, 231
curMlr
    tMPEG2PROGRAM, 218

decayCount
    tSTREAMSTATS, 250
description
    tALARMHANDLE, 196
destIpAddress
    tALIASCONFIG, 198
destIpMask
    tALIASCONFIG, 199
destMac
    tETHINFO, 205
destPort
    tALIASCONFIG, 199
detectedBitrate
    tCENTRY, 204
deviceRef
    tMPEG2PROGRAM, 218
dfAvg
    tMDIINFO, 215
dfCurrent
    tMDIINFO, 215
dfMax
    tMDIINFO, 215
dfMin
    tMDIINFO, 215
dfTotal
    tMDIINFO, 215
dstIP
    tIPINFO, 210
dstPort
    tIPINFO, 210
dupCurrent
    tRTPSTATS, 247
duplicate
    tMPEG2PID, 217

dupTotal
tRTPSTATS, 247

eFaults
tPMSTREAMGRAPHMETRICS, 235
eHistory
tPMSTREAMGRAPHMETRICS, 235
eMask
tPMSTREAMGRAPHMETRICS, 235
Ess
tPMSTREAMGRAPHMETRICS, 233
tPMSTREAMMETRICS, 237
tPMSTREAMTOTALMETRICS, 239
ess
tPMPIDSTATS, 225
tPMPROGRAMIVLSTATS, 227
tPMPROGRAMTOTSTATS, 231
eStateChanges
tPMSTREAMGRAPHMETRICS, 236
evtsP0
tPMSYSTEMMETRICS, 244
evtsP1
tPMSYSTEMMETRICS, 244
evtsP2
tPMSYSTEMMETRICS, 245
extFlags
tCENTRY, 204
tMPEG2PID, 217
tPMSTREAMGRAPHMETRICS, 235

faultHistory
tSTREAMSTATS, 250
faultMap
tSTREAMSTATS, 250
faults
tPMSTREAMMETRICS, 237
faultStatus
tPMSTREAMGRAPHMETRICS, 234
tSTREAMSTATS, 250
faultTime
tPMSTREAMGRAPHMETRICS, 234
tSTREAMSTATS, 250
fieldMask
tALIASCONFIG, 199
firstPidIndex
tMPEG2PROGRAM, 218
flags
tCENTRY, 204
tIGMPSTATS, 208
tMPEG2PID, 217
tMPEG2PROGRAM, 218
tPMSTREAMGRAPHMETRICS, 235
tPMSTREAMMETRICS, 237
tPMSYSTEMMETRICS, 245
tTUNERSTATS, 265
flowPayldStatus
tPMSTREAMGRAPHMETRICS, 236
flowStateCount
tPMSTREAMTOTALMETRICS, 240
fltMapChanged
tPMSYSTEMMETRICS, 243
fltPrograms
tPMSTREAMGRAPHMETRICS, 235
fltTsPids
tPMSTREAMGRAPHMETRICS, 235
fmTemplate
tALIASCONFIG, 199
freq
tTUNERRFMAP, 262
tTUNERSTATS, 263
future
tMTSPSTATS, 219

General Methods, 77
GENERALmeth
iqtcl_GetAPIVersionString, 80
iqtcl_GetCoreMode, 82
iqtcl_GetFirmwareMode, 85
iqtcl_GetLastError, 79
iqtcl_GetMACAddress, 81
iqtcl_GetTargetInfo, 86
iqtcl_GetTargetType, 84
iqtcl_GetTotalAvailableMemory, 87
iqtcl_SendCustomCommand, 90
iqtcl_SendSyslogMsg, 88
iqtcl_SendTargetSyslogMsg, 89
iqtcl_SetCoreMode, 83
iqtcl_SetDefaultTimeout, 78
greenStreams
tPMSYSTEMMETRICS, 243
greyStreams
tPMSYSTEMMETRICS, 243

handle
tALARMHANDLE, 196
tIGMPEVENT, 206
tIGMPSTATS, 208
hdrSize
tCENTRY, 204
hostIP
tASISTATUS, 203
tTAPSTATUS, 253

ID
tCENTRY, 204
Id
tALARMINFO, 197
id

tALIASCONFIG, 198
tASISTATUS, 203
tTAPSTATUS, 253
tTUNERSTATS, 264
IGMP
iqtcl_ClearIGMP, 105
iqtcl_GetIGMPFirst, 99
iqtcl_GetIGMPNext, 100
iqtcl_GetIGMPStatus, 98
iqtcl_IGMPJoin, 101
iqtcl_IGMPLeave, 102
iqtcl_StartIGMPLoop, 103
iqtcl_StopIGMPLoop, 104
IGMP Methods, 97
igmpSets
tALIASCONFIG, 201
igmpStatus
tALIASCONFIG, 199
intendedBitrate
tALIASCONFIG, 200
intendedType
tALIASCONFIG, 200
ipLoss
tPMSYSTEMMETRICS, 244
iqtcl_AddArpProxy
ARPPROXY, 92
iqtcl_AddArpProxyEx
ARPPROXY, 95
iqtcl_AddTriggerCondition
RECORDmeth, 149
iqtcl_ASILockStream
ASI, 39
iqtcl_ASIUnlockStream
ASI, 40
iqtcl_ClearAlarmLog
ALARM, 24
iqtcl_ClearAlarms
ALARM, 23
iqtcl_ClearAliases
ALIAS, 34
iqtcl_ClearArpProxyTable
ARPPROXY, 94
iqtcl_ClearCensus
CENSUS, 55
iqtcl_ClearIGMP
IGMP, 105
iqtcl_ClearPortCounters
PORT, 161
iqtcl_ClearSTBDefines
STB, 111
iqtcl_ClearStream
CENSUS, 56
iqtcl_ClearStreamStats
CENSUS, 57
iqtcl_ClearTracer
STIMULUS, 182
iqtcl_CloseCapture
CAPTURE, 44
iqtcl_CloseConnection
CONNECTION, 70
iqtcl_CloseOldConnections
CONNECTION, 74
iqtcl_CloseRecord
RECORDmeth, 142
iqtcl_CloseSession
CONNECTION, 72
iqtcl_CloseSTBSession
STB, 108
iqtcl_CloseStimulus
STIMULUS, 171
iqtcl_ConnectTo
CONNECTION, 67
iqtcl_DefineNewSTB
STB, 112
iqtcl_DownloadLibpcapFile
STIMULUS, 176
iqtcl_DownloadStimulusFile
STIMULUS, 173
iqtcl_DropPid
STIM_MOD, 190
iqtcl_GetActiveAlarmFirst
ALARM, 25
iqtcl_GetActiveAlarmNext
ALARM, 26
iqtcl_GetActiveAlarmTable
ALARM, 27
iqtcl_GetAlarmLogFirst
ALARM, 28
iqtcl_GetAlarmLogNext
ALARM, 29
iqtcl_GetAlarmLogTable
ALARM, 30
iqtcl_GetAliasFirst
ALIAS, 32
iqtcl_GetAliasNext
ALIAS, 33
iqtcl_GetAPIVersionString
GENERALmeth, 80
iqtcl_GetASIStatus
ASI, 38
iqtcl_GetCaptureStatus
CAPTURE, 48
iqtcl_GetCensusByID
CENSUS, 52
iqtcl_GetCensusFirst
CENSUS, 50
iqtcl_GetCensusNext
CENSUS, 51

iqtcl_GetCensusTable
CENSUS, 63
iqtcl_GetCensusTableItem
CENSUS, 58
iqtcl_GetCensusTableItemTagged
CENSUS, 59
iqtcl_GetCoreMode
GENERALmeth, 82
iqtcl_GetFirmwareMode
GENERALmeth, 85
iqtcl_GetIGMPFirst
IGMP, 99
iqtcl_GetIGMPNext
IGMP, 100
iqtcl_GetIGMPStatus
IGMP, 98
iqtcl_GetLastError
GENERALmeth, 79
iqtcl_GetLicenseCount
LICENSE, 116
iqtcl_GetMACAddress
GENERALmeth, 81
iqtcl_GetMDIStatus
MDI, 120
iqtcl_GetNextCensusID
CENSUS, 53
iqtcl_GetPidTableTagged
CENSUS, 65
iqtcl_GetPortCounterByIndex
PORT, 164
iqtcl_GetPortCounters
PORT, 163
iqtcl_GetPortCounterTable
PORT, 166
iqtcl_GetPortCounterTableItem
PORT, 165
iqtcl_GetPortStatus
PORT, 162
iqtcl_GetProgramTableTagged
CENSUS, 64
iqtcl_GetQAMTable
CENSUS, 62
iqtcl_GetQAMTableItem
CENSUS, 60
iqtcl_GetQAMTableItemTagged
CENSUS, 61
iqtcl_GetRecordStatus
RECORDmeth, 147
iqtcl_GetRVLStatus
RVL, 157
iqtcl_GetSTBDataTable
STB, 113
iqtcl_GetStimDiscovery
STIMULUS, 179
iqtcl_GetStimulusStatus
STIMULUS, 168
iqtcl_GetStreamID
CENSUS, 54
iqtcl_GetTargetInfo
GENERALmeth, 86
iqtcl_GetTargetType
GENERALmeth, 84
iqtcl_GetTotalAvailableMemory
GENERALmeth, 87
iqtcl_GetTriggerStatus
RECORDmeth, 151
iqtcl_IGMPJoin
IGMP, 101
iqtcl_IGMPLeave
IGMP, 102
iqtcl_IsConnected
CONNECTION, 71
iqtcl_IsLicenseValid
LICENSE, 115
iqtcl_IsQAMScanning
QAMmeth, 130
iqtcl_IsRecordTriggered
RECORDmeth, 152
iqtcl_LoadDefaultFile
STIMULUS, 174
iqtcl_MDILockStream
MDI, 121
iqtcl_MDIUnlockStream
MDI, 122
iqtcl_OpenCapture
CAPTURE, 42
iqtcl_OpenConnection
CONNECTION, 68
iqtcl_OpenConnectionWithMsgSize
CONNECTION, 69
iqtcl_OpenRecord
RECORDmeth, 140
iqtcl_OpenSmallStimulus
STIMULUS, 170
iqtcl_OpenSTBSession
STB, 107
iqtcl_OpenStimulus
STIMULUS, 169
iqtcl_OpenStreamCapture
CAPTURE, 43
iqtcl_OpenStreamRecord
RECORDmeth, 141
iqtcl_QAMGenerateAliasesFromLearn
QAMmeth, 127
iqtcl_QAMGetActiveChannel
QAMmeth, 125
iqtcl_QAMGetState
QAMmeth, 124

iqtcl_QAMGetStreamID
QAMmeth, 126
iqtcl_QAMStartLearn
QAMmeth, 131
iqtcl_QAMStartOp
QAMmeth, 132
iqtcl_QAMStartScan
QAMmeth, 129
iqtcl_QAMStartStream
QAMmeth, 128
iqtcl_QAMStopLearn
QAMmeth, 137
iqtcl_QAMStopOp
QAMmeth, 138
iqtcl_QAMStopScan
QAMmeth, 136
iqtcl_QAMStopStream
QAMmeth, 135
iqtcl_QAMTuneAndStreamBySTBChanName
QAMmeth, 133
iqtcl_QAMTuneAndStreamBySTBChanNumber
QAMmeth, 134
iqtcl_RemoveArpProxy
ARPPROXY, 93
iqtcl_RemoveArpProxyEx
ARPPROXY, 96
iqtcl_RemoveTriggerCondition
RECORDmeth, 150
iqtcl_ReplicateStream
STIMULUS, 172
iqtcl_ResumeSession
CONNECTION, 73
iqtcl_RetrieveSessionData
CONNECTION, 76
iqtcl_RVLLockStream
RVL, 158
iqtcl_RVLUnlockStream
RVL, 159
iqtcl_SaveSessionData
CONNECTION, 75
iqtcl_SaveUploadRecordToFile
RECORDmeth, 146
iqtcl_SendCustomCommand
GENERALmeth, 90
iqtcl_SendSyslogMsg
GENERALmeth, 88
iqtcl_SendTargetSyslogMsg
GENERALmeth, 89
iqtcl_SetBackgroundTraffic
STIMULUS, 175
iqtcl_SetBitrate
STIM_MOD, 189
iqtcl_SetCoreMode
GENERALmeth, 83
iqtcl_SetDefaultTimeout
GENERALmeth, 78
iqtcl_SetDFJitter
STIM_MOD, 187
iqtcl_SetIPDrops
STIM_MOD, 185
iqtcl_SetIPJitter
STIM_MOD, 186
iqtcl_SetPCRBitrate
STIM_MOD, 188
iqtcl_SetTracer
STIMULUS, 181
iqtcl_SetTriggerPos
RECORDmeth, 148
iqtcl_SetXCount
STIM_MOD, 184
iqtcl_StartASIScan
ASI, 36
iqtcl_StartCapture
CAPTURE, 45
iqtcl_StartIGMPLoop
IGMP, 103
iqtcl_StartMDIScan
MDI, 118
iqtcl_StartRecord
RECORDmeth, 143
iqtcl_StartRecord_TriggerPos
RECORDmeth, 153
iqtcl_StartRVLScan
RVL, 155
iqtcl_StartSTBSession
STB, 109
iqtcl_StartStimulus
STIMULUS, 177
iqtcl_StopASIScan
ASI, 37
iqtcl_StopCapture
CAPTURE, 46
iqtcl_StopDFJitter
STIM_MOD, 193
iqtcl_StopDropPid
STIM_MOD, 194
iqtcl_StopIGMPLoop
IGMP, 104
iqtcl_StopIPDrops
STIM_MOD, 191
iqtcl_StopIPJitter
STIM_MOD, 192
iqtcl_StopMDIScan
MDI, 119
iqtcl_StopRecord
RECORDmeth, 144
iqtcl_StopRVLScan
RVL, 156

iqtcl_StopSTBSession
  STB, 110
iqtcl_StopStimDiscovery
  STIMULUS, 180
iqtcl_StopStimulus
  STIMULUS, 178
iqtcl_UploadCapture
  CAPTURE, 47
iqtcl_UploadRecord
  RECORDmeth, 145
ivlFaults
  tPMPIDSTATS, 225
  tPMPROGRAMIVLSTATS, 228
ivlFlags
  tPMPROGRAMIVLSTATS, 229
ivlHist
  tPMPIDSTATS, 225
  tPMPROGRAMIVLSTATS, 228
ivlMask
  tPMPIDSTATS, 225
  tPMPROGRAMIVLSTATS, 228
ivlState
  tPMPIDSTATS, 225
ivlTime
  tPMSTREAMGRAPHMETRICS, 233
  tPMSYSTEMMETRICS, 242

join
  tIGMPEVENT, 206

language
  tMPEG2PID, 217
lastFlowPayldStatus
  tPMSTREAMGRAPHMETRICS, 234
lastLveTime
  tIGMPSTATS, 208
lastProgStatus
  tPMPROGRAMIVLSTATS, 229
lastTime
  tIGMPEVENT, 206
  tIGMPSTATS, 208
lbrAvg
  tSTREAMSTATS, 249
lbrCurrent
  tSTREAMSTATS, 249
lbrMax
  tSTREAMSTATS, 249
lbrMin
  tSTREAMSTATS, 249
lbrTotal
  tSTREAMSTATS, 249
ldCurrent
  tRTPSTATS, 247
ldErrors
  tRTPSTATS, 247
ldMin
  tRTPSTATS, 247
leave
  tIGMPEVENT, 206
ledToFaultMap
  tPMSTREAMTOTALMETRICS, 240
LICENSE
  iqtcl_GetLicenseCount, 116
  iqtcl_IsLicenseValid, 115
License Methods, 114
loss
  tPMPIDSTATS, 225
lossAlarms
  tPMPROGRAMIVLSTATS, 229
lossDuration
  tRTPSTATS, 247
lossEvtCurrent
  tRTPSTATS, 247
lossEvtTotal
  tRTPSTATS, 247
lossPercent
  tPMSTREAMGRAPHMETRICS, 236
  tPMSTREAMTOTALMETRICS, 240
  tRTPSTATS, 247
lossProgCount
  tPMSTREAMGRAPHMETRICS, 235
lossRatio
  tMPEG2PID, 217
lpCurrent
  tRTPSTATS, 247
lpErrors
  tPMSYSTEMMETRICS, 244
  tRTPSTATS, 247
lpMax
  tRTPSTATS, 247
ls24
  tPMSTREAMTOTALMETRICS, 240

mac
  tALIASCONFIG, 199
maxBitRate
  tPMSTREAMGRAPHMETRICS, 233
maxBr
  tPMPIDSTATS, 225
  tPMPROGRAMIVLSTATS, 227
maxLp
  tPMSYSTEMMETRICS, 244
maxMLp
  tPMPROGRAMIVLSTATS, 230
  tPMPROGRAMTOTSTATS, 231
maxMlr
  tPMPROGRAMIVLSTATS, 228
maxPktRate

tPMSTREAMGRAPHMETRICS, 234
maxTime
tIGMPEVENT, 206
tIGMPSTATS, 208
maxVBuffer
tPMSTREAMGRAPHMETRICS, 234
maxVTsb
tPMSTREAMGRAPHMETRICS, 234
mbrAvg
tSTREAMSTATS, 249
mbrCurrent
tSTREAMSTATS, 249
mbrMax
tSTREAMSTATS, 249
mbrMin
tSTREAMSTATS, 249
mbrTotal
tSTREAMSTATS, 249
MDI
iqtcl_GetMDIStatus, 120
iqtcl_MDILockStream, 121
iqtcl_MDIUnlockStream, 122
iqtcl_StartMDIScan, 118
iqtcl_StopMDIScan, 119
MDI Methods, 117
mdiDf
tPMSTREAMGRAPHMETRICS, 233
mediaLoss
tPMSYSTEMMETRICS, 243
mgtId
tPMSTREAMTOTALMETRICS, 240
minBitRate
tPMSTREAMGRAPHMETRICS, 233
minBr
tPMPIDSTATS, 225
tPMPROGRAMIVLSTATS, 227
minLd
tPMSYSTEMMETRICS, 244
minLp
tPMSYSTEMMETRICS, 244
minMLd
tPMPROGRAMIVLSTATS, 230
minPktRate
tPMSTREAMGRAPHMETRICS, 234
minTime
tIGMPEVENT, 206
tIGMPSTATS, 208
minVBuffer
tPMSTREAMGRAPHMETRICS, 234
minVTsb
tPMSTREAMGRAPHMETRICS, 234
misc
tMPEG2PID, 217
ml15
tMDIINFO, 215
ml24
tMDIINFO, 215
mlAvg
tMDIINFO, 215
mlCurrent
tMDIINFO, 215
mlMax
tMDIINFO, 215
mlMin
tMDIINFO, 215
Mls
tPMSTREAMGRAPHMETRICS, 233
tPMSTREAMMETRICS, 237
tPMSTREAMTOTALMETRICS, 239
mls
tPMSYSTEMMETRICS, 244
mls15
tPMPROGRAMIVLSTATS, 228
Mls24
tPMSTREAMTOTALMETRICS, 239
mls24
tPMPROGRAMTOTSTATS, 231
mlt15
tPMPROGRAMIVLSTATS, 228
mlt24
tPMPIDTOTALSTATS, 226
tPMPROGRAMTOTSTATS, 231
mlTotal
tMDIINFO, 215
mod
tTUNERSTATS, 264
modType
tALIASCONFIG, 199
monitors
tPMPROGRAMIVLSTATS, 229
monOutPids
tPMPROGRAMIVLSTATS, 229
monPids
tMTSPSTATS, 219
tPMPROGRAMIVLSTATS, 228
monPrograms
tPMSTREAMGRAPHMETRICS, 235
monTsPids
tPMSTREAMGRAPHMETRICS, 235
mtspSize
tMPEG2INFO, 216

name
tALIASCONFIG, 199
tMPEG2PROGRAM, 218
tNAMETAG, 220
nChannel
tMPEG2PROGRAM, 218

tOldMPEG2PROGRAM, 224
networkPid
tMPEG2INFO, 216
nFail
tIGMPSTATS, 208
nPids
tMPEG2PROGRAM, 218
tOldMPEG2PROGRAM, 224
nProgramAdded
tMPEG2INFO, 216
nProgramRemoved
tMPEG2INFO, 216
nPrograms
tMPEG2INFO, 216
nSamples
tMDIINFO, 215
tMPEG2PID, 217
tMTSPSTATS, 219
tSTREAMSTATS, 248
tTUNERSTATS, 264
nSamplesBer
tTUNERSTATS, 266
nSamplesRxPwr
tTUNERSTATS, 265
nSuccess
tIGMPSTATS, 208
num
tTUNERRFMAP, 262

oosCurrent
tRTPSTATS, 247
oosTotal
tRTPSTATS, 247
orangeStreams
tPMSYSTEMMETRICS, 243
outageCt
tPMSTREAMTOTALMETRICS, 239
outagePd
tMPEG2PID, 217
tPMPIDSTATS, 225
tPMPROGRAMIVLSTATS, 229
tPMSTREAMGRAPHMETRICS, 234
tPMSTREAMTOTALMETRICS, 239
outages
tMPEG2PID, 217
tPMPIDSTATS, 225
tPMPROGRAMIVLSTATS, 229
outPids
tPMPROGRAMIVLSTATS, 228

patVersion
tMPEG2INFO, 216
payloadSize
tCENTRY, 204
payloadType
tRTPINFO, 246
pbrAvg
tMPEG2PID, 217
tMTSPSTATS, 219
pbrCurrent
tMPEG2PID, 217
tMTSPSTATS, 219
pbrMax
tMPEG2PID, 217
tMTSPSTATS, 219
pbrMin
tMPEG2PID, 217
tMTSPSTATS, 219
pbrTotal
tMPEG2PID, 217
tMTSPSTATS, 219
pcrPid
tOldMPEG2PROGRAM, 224
Pess
tPMSTREAMGRAPHMETRICS, 233
tPMSTREAMTOTALMETRICS, 239
pid
tMPEG2PID, 217
pidStateCount
tPMPROGRAMTOTSTATS, 231
pktAvg
tSTREAMSTATS, 251
pktCurrent
tSTREAMSTATS, 251
pktLoss
tPMSTREAMGRAPHMETRICS, 233
tPMSTREAMTOTALMETRICS, 239
pktMax
tSTREAMSTATS, 251
pktMin
tSTREAMSTATS, 251
pktSizeMax
tSTREAMSTATS, 248
pktSizeMin
tSTREAMSTATS, 248
pktTotal
tSTREAMSTATS, 251
pmtPid
tOldMPEG2PROGRAM, 224
PORT
iqtcl_ClearPortCounters, 161
iqtcl_GetPortCounterByIndex, 164
iqtcl_GetPortCounters, 163
iqtcl_GetPortCounterTable, 166
iqtcl_GetPortCounterTableItem, 165
iqtcl_GetPortStatus, 162
Port Methods, 160
ports

tALIASCONFIG, 201
progAliases
tPMSTREAMTOTALMETRICS, 238
progNoAliasCnt
tPMSTREAMTOTALMETRICS, 238
progStateCount
tPMSTREAMTOTALMETRICS, 240
progStatus
tMPEG2PROGRAM, 218
tPMPROGRAMIVLSTATS, 228
protocol
tIPINFO, 210
providerName
tMPEG2PROGRAM, 218

QAM Methods, 123
QAMmeth
iqtcl_IsQAMScanning, 130
iqtcl_QAMGenerateAliasesFromLearn, 127
iqtcl_QAMGetActiveChannel, 125
iqtcl_QAMGetState, 124
iqtcl_QAMGetStreamID, 126
iqtcl_QAMStartLearn, 131
iqtcl_QAMStartOp, 132
iqtcl_QAMStartScan, 129
iqtcl_QAMStartStream, 128
iqtcl_QAMStopLearn, 137
iqtcl_QAMStopOp, 138
iqtcl_QAMStopScan, 136
iqtcl_QAMStopStream, 135
iqtcl_QAMTuneAndStreamBySTBChanName, 133
iqtcl_QAMTuneAndStreamBySTBChanNumber, 134

Record Methods, 139
RECORDmeth
iqtcl_AddTriggerCondition, 149
iqtcl_CloseRecord, 142
iqtcl_GetRecordStatus, 147
iqtcl_GetTriggerStatus, 151
iqtcl_IsRecordTriggered, 152
iqtcl_OpenRecord, 140
iqtcl_OpenStreamRecord, 141
iqtcl_RemoveTriggerCondition, 150
iqtcl_SaveUploadRecordToFile, 146
iqtcl_SetTriggerPos, 148
iqtcl_StartRecord, 143
iqtcl_StartRecord_TriggerPos, 153
iqtcl_StopRecord, 144
iqtcl_UploadRecord, 145
redStreams
tPMSYSTEMMETRICS, 243
result
tIGMPEVENT, 206
retryFills
tPMSTREAMGRAPHMETRICS, 236
tPMSYSTEMMETRICS, 244
retryReqs
tPMSTREAMGRAPHMETRICS, 236
tPMSYSTEMMETRICS, 244
rsBytesTotal
tTUNERSTATS, 265
rsCoCurrent
tTUNERSTATS, 265
rsCoPercent
tTUNERSTATS, 265
rsCoTotal
tTUNERSTATS, 265
rsUcCurrent
tTUNERSTATS, 264
rsUcPercent
tTUNERSTATS, 265
rsUcTotal
tTUNERSTATS, 265
rtpLd
tPMSTREAMGRAPHMETRICS, 233
rtpLoss
tPMSTREAMGRAPHMETRICS, 235
rtpLoss24
tPMSTREAMTOTALMETRICS, 240
rtpLossEvts
tPMSTREAMGRAPHMETRICS, 236
rtplP
tPMSTREAMGRAPHMETRICS, 236
rtpLs
tPMSTREAMGRAPHMETRICS, 236
RVL
iqtcl_GetRVLStatus, 157
iqtcl_RVLLockStream, 158
iqtcl_RVLUnlockStream, 159
iqtcl_StartRVLScan, 155
iqtcl_StopRVLScan, 156
RVL Methods, 154
RxAlignErrors
tTritenCtrs, 257
RxBadFrames
tIXF18103CTRS, 214
RxBroadcastFrames
tIXF18103CTRS, 212
RxCrcError
tNemoCtrs, 222
RxDataErrors
tTritenCtrs, 257
RxErrors
tNemoCtrs, 222
RxEthFcs
tIXF18103CTRS, 213

RxEthFragments
tIXF18103CTRS, 213
RxEthJabbers
tIXF18103CTRS, 213
RxEthOctets
tIXF18103CTRS, 213
RxEthOversized
tIXF18103CTRS, 213
RxEthPkts
tIXF18103CTRS, 213
RxEthUndersized
tIXF18103CTRS, 213
RxFcsErrors
tTritenCtrs, 257
RxGoodFrames
tIXF18103CTRS, 214
RxJabberErrors
tTritenCtrs, 257
RxLongErrors
tTritenCtrs, 257
RxMACCtrlFrames
tIXF18103CTRS, 213
RxMulticastFrames
tIXF18103CTRS, 212
RxOctetsBad
tTritenCtrs, 257
RxOctetsGood
tTritenCtrs, 257
RxOctetsTotal
tNemoCtrs, 221
RxPackets
tNemoCtrs, 221
tTritenCtrs, 257
RxPacketsBroadcast
tNemoCtrs, 221
tTritenCtrs, 257
RxPacketsMulticast
tNemoCtrs, 221
tTritenCtrs, 257
RxPacketsTotal
tNemoCtrs, 221
RxPacketsUnicast
tNemoCtrs, 221
tTritenCtrs, 257
RxPauseControl
tTritenCtrs, 258
RxPAUSECtrlFrames
tIXF18103CTRS, 213
rxPwrAvg
tTUNERSTATS, 266
rxPwrCurrent
tTUNERSTATS, 266
rxPwrMax
tTUNERSTATS, 266
rxPwrMin
tTUNERSTATS, 266
rxPwrTotal
tTUNERSTATS, 266
RxRuntErrors
tTritenCtrs, 258
RxSequenceErrors
tTritenCtrs, 258
RxShortErrors
tTritenCtrs, 258
RxSizeFrames
tIXF18103CTRS, 212
RxSymbolErrors
tTritenCtrs, 258
RxTagged
tTritenCtrs, 257
RxTotalBytes
tIXF18103CTRS, 214
RxTotalErrors
tTritenCtrs, 258
RxTotalFrames
tIXF18103CTRS, 212
RxTotalOctets
tIXF18103CTRS, 212
tTritenCtrs, 258
RxTotalPackets
tTritenCtrs, 258
RxUnicastFrames
tIXF18103CTRS, 213
RxUnknownControl
tTritenCtrs, 258
RxVeryLongErrors
tTritenCtrs, 258
RxVLANFrames
tIXF18103CTRS, 212

scteEvtTime
tPMPROGRAMTOTSTATS, 231
Sdps
tPMSTREAMMETRICS, 237
seqErrCurrent
tRTPSTATS, 247
seqErrTotal
tRTPSTATS, 247
Sess
tPMSTREAMGRAPHMETRICS, 233
tPMSTREAMMETRICS, 237
tPMSTREAMTOTALMETRICS, 239
severity
tALARMINFO, 197
signal
tTUNERSTATS, 264
size
tTAG, 252

snrAvg
tTUNERSTATS, 264
snrCurrent
tTUNERSTATS, 264
snrMax
tTUNERSTATS, 264
snrMin
tTUNERSTATS, 264
snrTotal
tTUNERSTATS, 264
srcFilter
tIGMPEVENT, 206
tIGMPSTATS, 208
srcIP
tIPINFO, 210
srcIpAddress
tALIASCONFIG, 198
srcIpMask
tALIASCONFIG, 199
srcMac
tETHINFO, 205
srcPort
tALIASCONFIG, 199
tIPINFO, 210
ssrc
tALIASCONFIG, 200
starts
tPMSTREAMGRAPHMETRICS, 234
tPMSTREAMMETRICS, 237
state
tIGMPEVENT, 206
stateChanges
tPMPROGRAMIVLSTATS, 230
stateCount
tPMPROGRAMTOTSTATS, 231
stateFlags
tPMPIDTOTALSTATS, 226
stateTime
tMPEG2PID, 217
tMPEG2PROGRAM, 218
tPMSTREAMTOTALMETRICS, 240
status
tALARMINFO, 197
tASISTATUS, 203
tTAPSTATUS, 253
STB
iqtcl_ClearSTBDefines, 111
iqtcl_CloseSTBSession, 108
iqtcl_DefineNewSTB, 112
iqtcl_GetSTBDataTable, 113
iqtcl_OpenSTBSession, 107
iqtcl_StartSTBSession, 109
iqtcl_StopSTBSession, 110
STB Methods, 106
STIM_MOD
iqtcl_DropPid, 190
iqtcl_SetBitrate, 189
iqtcl_SetDFJitter, 187
iqtcl_SetIPDrops, 185
iqtcl_SetIPJitter, 186
iqtcl_SetPCRBitrate, 188
iqtcl_SetXCount, 184
iqtcl_StopDFJitter, 193
iqtcl_StopDropPid, 194
iqtcl_StopIPDrops, 191
iqtcl_StopIPJitter, 192
STIMULUS
iqtcl_ClearTracer, 182
iqtcl_CloseStimulus, 171
iqtcl_DownloadLibpcapFile, 176
iqtcl_DownloadStimulusFile, 173
iqtcl_GetStimDiscovery, 179
iqtcl_GetStimulusStatus, 168
iqtcl_LoadDefaultFile, 174
iqtcl_OpenSmallStimulus, 170
iqtcl_OpenStimulus, 169
iqtcl_ReplicateStream, 172
iqtcl_SetBackgroundTraffic, 175
iqtcl_SetTracer, 181
iqtcl_StartStimulus, 177
iqtcl_StopStimDiscovery, 180
iqtcl_StopStimulus, 178
Stimulus Methods, 167
Stimulus Modification Methods, 183
stopTime
tPMPIDTOTALSTATS, 226
streamId
tALARMINFO, 197
tPMSTREAMGRAPHMETRICS, 233
tPMSTREAMMETRICS, 237
tPMSTREAMTOTALMETRICS, 238
streamType
tCENTRY, 204
symbolRate
tTUNERRFMAP, 262
syncError
tMTSPSTATS, 219
syncErrorTotal
tMTSPSTATS, 219
systemStatus
tPMSYSTEMMETRICS, 245

tag
tALARMHANDLE, 196
tALARMINFO, 197
tALIASCONFIG, 198
tALIASNAME, 202
tASISTATUS, 203

tCENTRY, 204
tETHINFO, 205
tIGMPEVENT, 206
tIGMPGROUPS, 207
tIGMPSTATS, 208
tIGMPSTATUS, 209
tIPINFO, 210
tIXF18103CTRS, 211
tMDIINFO, 215
tMPEG2INFO, 216
tMPEG2PID, 217
tMPEG2PROGRAM, 218
tMTSPSTATS, 219
tNAMETAG, 220
tNEMOCTRS, 223
tOldMPEG2PROGRAM, 224
tPMPIDSTATS, 225
tPMPIDTOTALSTATS, 226
tPMPROGRAMIVLSTATS, 227
tPMPROGRAMTOTSTATS, 231
tPMSTREAMGRAPHMETRICS, 233
tPMSTREAMMETRICS, 237
tPMSTREAMTOTALMETRICS, 238
tPMSYSTEMMETRICS, 242
tRTPINFO, 246
tRTPSTATS, 247
tSTREAMSTATS, 248
tTAPSTATUS, 253
tTARGETINFO, 254
tTRITENCTRS, 255
tTUNERRFMAP, 262
tTUNERSTATS, 263
tALARMHANDLE, 196
description, 196
handle, 196
tag, 196
timestamp, 196
tALARMINFO, 197
alarmId, 197
Id, 197
severity, 197
status, 197
streamId, 197
tag, 197
threshold, 197
timestamp, 197
value, 197
tALIASCONFIG, 198
aliasType, 200
bJoined, 199
charTemplate, 200
configStatus, 200
destIpAddress, 198
destIpMask, 199
destPort, 199
fieldMask, 199
fmTemplate, 199
id, 198
igmpSets, 201
igmpStatus, 199
intendedBitrate, 200
intendedType, 200
mac, 199
modType, 199
name, 199
ports, 201
srcIpAddress, 198
srcIpMask, 199
srcPort, 199
ssrc, 200
tag, 198
tsId, 201
tunerSdvDesc, 200
tunerSdvMaxBw, 200
tunerSdvType, 200
videoType, 200
vlanTci, 200
tALIASNAME, 202
alias, 202
tag, 202
TapBytesDropped
tTritenCtrs, 261
TapPacketsDropped
tTritenCtrs, 261
targetContact
tTARGETINFO, 254
targetLocation
tTARGETINFO, 254
targetMode
tTARGETINFO, 254
targetName
tTARGETINFO, 254
tASISTATUS, 203
hostIP, 203
id, 203
status, 203
tag, 203
taskStatus
tIGMPSTATUS, 209
tBadStreams
tPMSYSTEMMETRICS, 242
tCENTRY, 204
bitrate, 204
detectedBitrate, 204
extFlags, 204
flags, 204
hdrSize, 204
ID, 204

payloadSize, 204
streamType, 204
tag, 204
timestamp, 204
tETHINFO, 205
destMac, 205
srcMac, 205
tag, 205
threshold
tALARMINFO, 197
tIGMPEVENT, 206
address, 206
avgTime, 206
handle, 206
join, 206
lastTime, 206
leave, 206
maxTime, 206
minTime, 206
result, 206
srcFilter, 206
state, 206
tag, 206
vlan, 206
tIGMPGROUPS, 207
address, 207
tag, 207
tIGMPSTATS, 208
address, 208
avgTime, 208
flags, 208
handle, 208
lastLveTime, 208
lastTime, 208
maxTime, 208
minTime, 208
nFail, 208
nSuccess, 208
srcFilter, 208
tag, 208
vlan, 208
tIGMPSTATUS, 209
tag, 209
taskStatus, 209
timeDate
tTARGETINFO, 254
timestamp
tALARMHANDLE, 196
tALARMINFO, 197
tCENTRY, 204
tPMSYSTEMMETRICS, 245
tIPINFO, 210
dstIP, 210
dstPort, 210
protocol, 210
srcIP, 210
srcPort, 210
tag, 210
tos, 210
vlanID, 210
tIXF18103CTRS, 211
RxBadFrames, 214
RxBroadcastFrames, 212
RxEthFcs, 213
RxEthFragments, 213
RxEthJabbers, 213
RxEthOctets, 213
RxEthOversized, 213
RxEthPkts, 213
RxEthUndersized, 213
RxGoodFrames, 214
RxMACCtrlFrames, 213
RxMulticastFrames, 212
RxPAUSECtrlFrames, 213
RxSizeFrames, 212
RxTotalBytes, 214
RxTotalFrames, 212
RxTotalOctets, 212
RxUnicastFrames, 213
RxVLANFrames, 212
tag, 211
TxBroadcastFrames, 211
TxMACCtrlFrames, 212
TxMulticastFrames, 211
TxPAUSECtrlFrames, 212
TxSizeFrames, 212
TxTotalBytes, 213
TxTotalFrames, 211
TxTotalOctets, 211
TxUnicastFrames, 212
TxVLANFrames, 212
tMaxStreams
tPMSYSTEMMETRICS, 243
tMDIINFO, 215
dfAvg, 215
dfCurrent, 215
dfMax, 215
dfMin, 215
dfTotal, 215
ml15, 215
ml24, 215
mlAvg, 215
mlCurrent, 215
mlMax, 215
mlMin, 215
mlTotal, 215
nSamples, 215
tag, 215

vbAvg, 215
vbCurrent, 215
vbMax, 215
vbMin, 215
vbTotal, 215
tMinStreams
tPMSYSTEMMETRICS, 243
tMPEG2INFO, 216
mtspSize, 216
networkPid, 216
nProgramAdded, 216
nProgramRemoved, 216
nPrograms, 216
patVersion, 216
tag, 216
tsId, 216
tMPEG2PID, 217
ccErrCurrent, 217
ccErrTotal, 217
duplicate, 217
extFlags, 217
flags, 217
language, 217
lossRatio, 217
misc, 217
nSamples, 217
outagePd, 217
outages, 217
pbrAvg, 217
pbrCurrent, 217
pbrMax, 217
pbrMin, 217
pbrTotal, 217
pid, 217
stateTime, 217
tag, 217
type, 217
tMPEG2PROGRAM, 218
alarmPids, 218
aliasName, 218
chanNumber, 218
crc, 218
curBitrate, 218
curMlr, 218
deviceRef, 218
firstPidIndex, 218
flags, 218
name, 218
nChannel, 218
nPids, 218
progStatus, 218
providerName, 218
stateTime, 218
tag, 218
totLoss, 218
tMTSPSTATS, 219
almPids, 219
ccErrCurrent, 219
ccErrTotal, 219
future, 219
monPids, 219
nSamples, 219
pbrAvg, 219
pbrCurrent, 219
pbrMax, 219
pbrMin, 219
pbrTotal, 219
syncError, 219
syncErrorTotal, 219
tag, 219
totalPids, 219
tNAMETAG, 220
name, 220
tag, 220
tNEMOCTRS, 223
ctrs, 223
tag, 223
tNemoCtrs, 221
RxCrcError, 222
RxErrors, 222
RxOctetsTotal, 221
RxPackets, 221
RxPacketsBroadcast, 221
RxPacketsMulticast, 221
RxPacketsTotal, 221
RxPacketsUnicast, 221
TxOctetsTotal, 221
TxPackets, 222
TxPacketsBroadcast, 222
TxPacketsMulticast, 222
TxPacketsTotal, 221
TxPacketsUnicast, 222
tNewStreams
tPMSYSTEMMETRICS, 242
tOldMPEG2PROGRAM, 224
nChannel, 224
nPids, 224
pcrPid, 224
pmtPid, 224
tag, 224
tos
tIPINFO, 210
tSTREAMSTATS, 250
totAlarms
tPMPIDTOTALSTATS, 226
totalBr
tPMPIDTOTALSTATS, 226
totalPids

tMTSPSTATS, 219
totEss
tPMPIDTOTALSTATS, 226
tPMPROGRAMIVLSTATS, 229
tPMPROGRAMTOTSTATS, 231
tPMSTREAMTOTALMETRICS, 241
totLoss
tMPEG2PROGRAM, 218
totMaxMLp
tPMPROGRAMTOTSTATS, 231
totMinMLd
tPMPROGRAMTOTSTATS, 231
totMloss
tPMSTREAMTOTALMETRICS, 240
totMLS
tPMPROGRAMTOTSTATS, 231
totMls
tPMPROGRAMIVLSTATS, 230
totMLT
tPMPROGRAMTOTSTATS, 231
totOutagePd
tPMPIDTOTALSTATS, 226
tPMPROGRAMIVLSTATS, 229
tPMPROGRAMTOTSTATS, 231
tPMSTREAMTOTALMETRICS, 240
totPess
tPMSTREAMTOTALMETRICS, 241
totScteEvts
tPMPROGRAMIVLSTATS, 229
tPMPROGRAMTOTSTATS, 231
Totss
tPMSTREAMTOTALMETRICS, 239
Totsts
tPMSTREAMTOTALMETRICS, 239
tPMPIDSTATS, 225
alarms, 225
avgBr, 225
ess, 225
ivlFaults, 225
ivlHist, 225
ivlMask, 225
ivlState, 225
loss, 225
maxBr, 225
minBr, 225
outagePd, 225
outages, 225
tag, 225
tPMPIDTOTALSTATS, 226
mlt24, 226
stateFlags, 226
stopTime, 226
tag, 226
totAlarms, 226
totalBr, 226
totEss, 226
totOutagePd, 226
tPMPROGRAMIVLSTATS, 227
alarms, 228
almPids, 228
avgBr, 229
ess, 227
ivlFaults, 228
ivlFlags, 229
ivlHist, 228
ivlMask, 228
lastProgStatus, 229
lossAlarms, 229
maxBr, 227
maxMLp, 230
maxMlr, 228
minBr, 227
minMLd, 230
mls15, 228
mlt15, 228
monitors, 229
monOutPids, 229
monPids, 228
outagePd, 229
outages, 229
outPids, 228
progStatus, 228
stateChanges, 230
tag, 227
totEss, 229
totMls, 230
totOutagePd, 229
totScteEvts, 229
tPMPROGRAMTOTSTATS, 231
curMLd, 231
curMLp, 231
ess, 231
maxMLp, 231
mls24, 231
mlt24, 231
pidStateCount, 231
scteEvtTime, 231
stateCount, 231
tag, 231
totEss, 231
totMaxMLp, 231
totMinMLd, 231
totMLS, 231
totMLT, 231
totOutagePd, 231
totScteEvts, 231
tPMSTREAMGRAPHMETRICS, 232
eFaults, 235

eHistory, 235
eMask, 235
Ess, 233
eStateChanges, 236
extFlags, 235
faultStatus, 234
faultTime, 234
flags, 235
flowPayldStatus, 236
fltPrograms, 235
fltTsPids, 235
ivlTime, 233
lastFlowPayldStatus, 234
lossPercent, 236
lossProgCount, 235
maxBitRate, 233
maxPktRate, 234
maxVBuffer, 234
maxVTsb, 234
mdiDf, 233
minBitRate, 233
minPktRate, 234
minVBuffer, 234
minVTsb, 234
Mls, 233
monPrograms, 235
monTsPids, 235
outagePd, 234
Pess, 233
pktLoss, 233
retryFills, 236
retryReqs, 236
rtpLd, 233
rtpLoss, 235
rtpLossEvts, 236
rtplP, 236
rtpLs, 236
Sess, 233
starts, 234
streamId, 233
tag, 233
usrFeebacks, 236

tPMSTREAMMETRICS, 237
Ess, 237
faults, 237
flags, 237
Mls, 237
Sdps, 237
Sess, 237
starts, 237
streamId, 237
tag, 237
Uass, 237

tPMSTREAMTOTALMETRICS, 238
Actss, 239
Ess, 239
flowStateCount, 240
ledToFaultMap, 240
lossPercent, 240
ls24, 240
mgtId, 240
Mls, 239
Mls24, 239
outageCt, 239
outagePd, 239
Pess, 239
pktLoss, 239
progAliases, 238
progNoAliasCnt, 238
progStateCount, 240
rtpLoss24, 240
Sess, 239
stateTime, 240
streamId, 238
tag, 238
totEss, 241
totMloss, 240
totOutagePd, 240
totPess, 241
Totss, 239
Totsts, 239
usrQos, 240

tPMSYSTEMMETRICS, 242
activeStreams, 243
bcastStreams, 245
blueStreams, 243
evtsP0, 244
evtsP1, 244
evtsP2, 245
flags, 245
fltMapChanged, 243
greenStreams, 243
greyStreams, 243
ipLoss, 244
ivlTime, 242
lpErrors, 244
maxLp, 244
mediaLoss, 243
minLd, 244
minLp, 244
mls, 244
orangeStreams, 243
redStreams, 243
retryFills, 244
retryReqs, 244
systemStatus, 245
tag, 242
tBadStreams, 242

timestamp, 245
tMaxStreams, 243
tMinStreams, 243
tNewStreams, 242
trapSentRate, 245
usrQos, 244
util, 243

trapSentRate
tPMSYSTEMMETRICS, 245

tRTPINFO, 246
payloadType, 246
tag, 246

tRTPSTATS, 247
dupCurrent, 247
dupTotal, 247
ldCurrent, 247
ldErrors, 247
ldMin, 247
lossDuration, 247
lossEvtCurrent, 247
lossEvtTotal, 247
lossPercent, 247
lpCurrent, 247
lpErrors, 247
lpMax, 247
oosCurrent, 247
oosTotal, 247
seqErrCurrent, 247
seqErrTotal, 247
tag, 247

tsId
tALIASCONFIG, 201
tMPEG2INFO, 216

tSTREAMSTATS, 248
decayCount, 250
faultHistory, 250
faultMap, 250
faultStatus, 250
faultTime, 250
lbrAvg, 249
lbrCurrent, 249
lbrMax, 249
lbrMin, 249
lbrTotal, 249
mbrAvg, 249
mbrCurrent, 249
mbrMax, 249
mbrMin, 249
mbrTotal, 249
nSamples, 248
pktAvg, 251
pktCurrent, 251
pktMax, 251
pktMin, 251
pktSizeMax, 248
pktSizeMin, 248
pktTotal, 251
tag, 248
tos, 250
userFeedback, 250
utilAvg, 250
utilCurrent, 250
utilMax, 250
utilMin, 249
utilTotal, 250

tTAG, 252
size, 252
type, 252

tTAPSTATUS, 253
hostIP, 253
id, 253
status, 253
tag, 253

tTARGETINFO, 254
tag, 254
targetContact, 254
targetLocation, 254
targetMode, 254
targetName, 254
timeDate, 254

tTRITENCTRS, 255
ctrs, 255
tag, 255

tTritenCtrs, 256
CarrierExtendErrors, 258
RxAlignErrors, 257
RxDataErrors, 257
RxFcsErrors, 257
RxJabberErrors, 257
RxLongErrors, 257
RxOctetsBad, 257
RxOctetsGood, 257
RxPackets, 257
RxPacketsBroadcast, 257
RxPacketsMulticast, 257
RxPacketsUnicast, 257
RxPauseControl, 258
RxRuntErrors, 258
RxSequenceErrors, 258
RxShortErrors, 258
RxSymbolErrors, 258
RxTagged, 257
RxTotalErrors, 258
RxTotalOctets, 258
RxTotalPackets, 258
RxUnknownControl, 258
RxVeryLongErrors, 258
TapBytesDropped, 261

TapPacketsDropped, 261
TxDeffered, 259
TxExcessiveCollisionErrors, 260
TxExcessiveDefferalErrors, 260
TxExcessiveLengthDrop, 260
TxFcsErrors, 260
TxFlowControlCollisions, 260
TxLateCollisions, 259
TxMultipleCollisions, 259
TxOctetsBad, 259
TxOctetsGood, 259
TxPackets, 259
TxPacketsBroadcast, 259
TxPacketsMulticast, 259
TxPacketsUnicast, 259
TxPauseFrames, 260
TxSingleCollisions, 259
TxTagged, 260
TxTotalCollisions, 259
TxTotalErrors, 260
TxTotalOctets, 260
TxTotalPackets, 260
TxUnderrun, 260
tTUNERRFMAP, 262
freq, 262
num, 262
symbolRate, 262
tag, 262
tTUNERSTATS, 263
active, 264
align, 265
berPostAvg, 267
berPostCurrent, 267
berPostMax, 267
berPostMin, 267
berPostTotal, 267
berPreAvg, 266
berPreCurrent, 266
berPreMax, 266
berPreMin, 266
berPreTotal, 266
chan, 263
chanPre, 265
flags, 265
freq, 263
id, 264
mod, 264
nSamples, 264
nSamplesBer, 266
nSamplesRxPwr, 265
rsBytesTotal, 265
rsCoCurrent, 265
rsCoPercent, 265
rsCoTotal, 265
rsUcCurrent, 264
rsUcPercent, 265
rsUcTotal, 265
rxPwrAvg, 266
rxPwrCurrent, 266
rxPwrMax, 266
rxPwrMin, 266
rxPwrTotal, 266
signal, 264
snrAvg, 264
snrCurrent, 264
snrMax, 264
snrMin, 264
snrTotal, 264
tag, 263
tuner, 265
tuner
tTUNERSTATS, 265
tunerSdvDesc
tALIASCONFIG, 200
tunerSdvMaxBw
tALIASCONFIG, 200
tunerSdvType
tALIASCONFIG, 200
TxBroadcastFrames
tIXF18103CTRS, 211
TxDeffered
tTritenCtrs, 259
TxExcessiveCollisionErrors
tTritenCtrs, 260
TxExcessiveDefferalErrors
tTritenCtrs, 260
TxExcessiveLengthDrop
tTritenCtrs, 260
TxFcsErrors
tTritenCtrs, 260
TxFlowControlCollisions
tTritenCtrs, 260
TxLateCollisions
tTritenCtrs, 259
TxMACCtrlFrames
tIXF18103CTRS, 212
TxMulticastFrames
tIXF18103CTRS, 211
TxMultipleCollisions
tTritenCtrs, 259
TxOctetsBad
tTritenCtrs, 259
TxOctetsGood
tTritenCtrs, 259
TxOctetsTotal
tNemoCtrs, 221
TxPackets
tNemoCtrs, 222

tTritenCtrs, 259
TxPacketsBroadcast
tNemoCtrs, 222
tTritenCtrs, 259
TxPacketsMulticast
tNemoCtrs, 222
tTritenCtrs, 259
TxPacketsTotal
tNemoCtrs, 221
TxPacketsUnicast
tNemoCtrs, 222
tTritenCtrs, 259
TxPAUSECtrlFrames
tIXF18103CTRS, 212
TxPauseFrames
tTritenCtrs, 260
TxSingleCollisions
tTritenCtrs, 259
TxSizeFrames
tIXF18103CTRS, 212
TxTagged
tTritenCtrs, 260
TxTotalBytes
tIXF18103CTRS, 213
TxTotalCollisions
tTritenCtrs, 259
TxTotalErrors
tTritenCtrs, 260
TxTotalFrames
tIXF18103CTRS, 211
TxTotalOctets
tIXF18103CTRS, 211
tTritenCtrs, 260
TxTotalPackets
tTritenCtrs, 260
TxUnderrun
tTritenCtrs, 260
TxUnicastFrames
tIXF18103CTRS, 212
TxVLANFrames
tIXF18103CTRS, 212
type
tMPEG2PID, 217
tTAG, 252

Uass
tPMSTREAMMETRICS, 237
userFeedback
tSTREAMSTATS, 250
usrFeebacks
tPMSTREAMGRAPHMETRICS, 236
usrQos
tPMSTREAMTOTALMETRICS, 240
tPMSYSTEMMETRICS, 244
util
tPMSYSTEMMETRICS, 243
utilAvg
tSTREAMSTATS, 250
utilCurrent
tSTREAMSTATS, 250
utilMax
tSTREAMSTATS, 250
utilMin
tSTREAMSTATS, 249
utilTotal
tSTREAMSTATS, 250

value
tALARMINFO, 197
vbAvg
tMDIINFO, 215
vbCurrent
tMDIINFO, 215
vbMax
tMDIINFO, 215
vbMin
tMDIINFO, 215
vbTotal
tMDIINFO, 215
videoType
tALIASCONFIG, 200
vlan
tIGMPEVENT, 206
tIGMPSTATS, 208
vlanID
tIPINFO, 210
vlanTci
tALIASCONFIG, 200