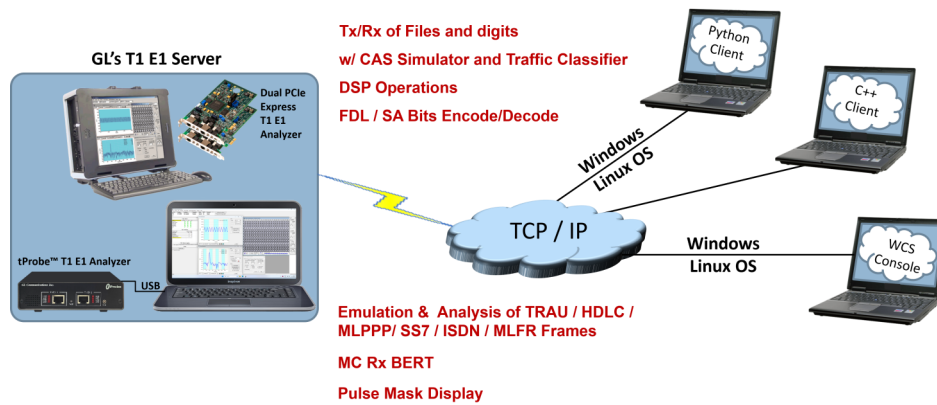


Windows Client / Server for T1 E1 Analyzer



Overview

GL's Windows Client / Server software provides the capability of remote operation, automation, and multi-site connectivity to the users over GL's T1 E1 analyzers. T1 E1 analyzers in a server mode can be easily controlled through software clients at remote or local sites via TCP/IP sockets. Connectivity can be via Dial-Up, LAN, WAN, or more typically the Internet. Server software runs multiple tasks simultaneously at the request of the client software. Server software runs under Windows® and Linux® operating system.

Windows Client Server (WCS) is available as a part of T1 E1 analyzer basic applications. However, with additional licenses, various modules obtained to suit the testing requirements. WCS software supports Python scripting on both Windows® and Linux® operating systems. Clients can be written in C++, C#, Windows Tool Command Language (TCL), and Windows/Linux Python (versions 3.6 or above). WCS clients are available via console/terminal Command Line Interface (CLI) and accessible remotely through SSH.

Typical applications include:

- Supports for framed and unframed transmission/ reception of signals
- Intrusive/Non-Intrusive T1 E1 Testing
- Performance monitoring and testing of multiple site locations from a single client
- Shared use of T1 E1 test equipment from multiple client locations
- Automated factory testing on production lines
- Integration of T1 E1 testing into more complex testing systems
- Examples of monitoring Tasks
 - Monitor Signaling Bits
 - Detect and report DTMF/MF/MFC-R2 digits on channels as they occur
 - Precision Delay Measurement
 - Traffic Classifier
 - Collection of call records from remote locations based on signaling (SS7, CAS, ISDN, etc.)
 - Monitor, report, and record alarms at various sites every two seconds or as they occur
- Record/Playback functions
 - File based Record/Playback
 - Client Data Tx Rx
- Intrusive Testing
 - Supports auto and manual error insertions along with Bulk delay to apply delay on the entire T1 or E1 trunk
 - Dual VF Tx Rx (earlier called as Simplified Audio Client) to perform analog Tx/Rx and Digital T1 testing functionalities
- Simple modifiable scripts to run each and every G.168 tests
- Perform BERT on selected timeslots involving multiple paths simultaneously

For more details, visit [Windows Client/Server for T1 E1 Analysis](#) webpage.



818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878, U.S.A
(Web) www.gl.com - (V) +1-301-670-4784 (F) +1-301-670-9187 - (E-Mail) info@gl.com

Basic WCS Operations

Dual VF Tx Rx

Dual VF Tx Rx (earlier called as Simplified Audio Client) is a client-server based application that provides an alternate and simple GUI to perform analog Tx/Rx and Digital T1 testing functionalities. SAC supports analog operational mode featuring two VF interfaces per unit, with each VF interface providing independent Tx/Rx functionality. It includes spectral displays and in-built speakers for analyzing analog signals. Dual VF Tx Rx (XX605) is available as a part of T1 E1 analyzer basic applications.

BERT

Provides commands that allow you to run full-frame or fractional bit error rate tests. Data rates of 64 kbps (64k) and 56 kbps (56k) are supported. The test is based on the transmission of a known pattern onto the specified transmit timeslots, and the receipt of that same pattern, perhaps with errors, on the corresponding receive timeslots.

Sample scripts for BERT

```
bert 64k pattern "qrss" #1 framed 1min;
//running "full framed qrss" bert pattern for 1 minute
```

Error Insertion Module

This application permits inserting single, fixed, automatic, random, and burst error into the incoming bit stream. Bulk delay feature helps to simulate network delay along the T1 E1 links Maximum CRC and Frame error rate that can be inserted in the Error Insertion (T1/ESF) application has been increased to 1:1 from 1:100.

Sample scripts for bulk delay insertion for USB or Universal only

```
run task "ErrInstnE1:InsertError" #1:* continuous;
inform task 1 "set txtimeslots 1-5,8,20";
inform task 1 "set bulkdelayunits milliseconds";
inform task 1 "set bulkdelaytime 87";
// set to 87 ms (milliseconds)
inform task 1 "set bulkdelay on";
inform task 1 "display";
```

Monitor Signaling Bits—WaitSigBits Module

WaitSigBits is a DLL that waits for a specified signaling bits pattern to appear on a specified timeslot before terminating.

Sample scripts for bulk delay insertion for USB or Universal only

```
run task "waitsigbits1" using "1,0,0,1" #1:1;
//Instructs the server to monitor signaling bits on port #1 timeslot 1
//for the signaling bit pattern 1,0,0,1. The task waits indefinitely
//for the signaling bits to arrive. The default response time is used.
```

Precision Delay Measurement Module

PrecDelayMeas is a DLL that performs precision delay measurement on a digital system.

Sample scripts for bulk delay insertion for USB or Universal only

```
run task "PrecDelayMeasE1" #1;
//Instructs the server to conduct the precision delay measurement function on card #1.
run task "PrecDelayMeasT1" using "56k" #2:0..11 10 sec;
//Instructs the server to conduct the precision delay measurement
//function on card #2, timeslots 0 through 11 inclusive. The task
//times out in 10 seconds if the measurement is unsuccessful.
```

Basic WCS Operations (Contd.)

Monitoring Alarms

Monitoring alarms is very similar to querying alarms, except that monitoring alarms is carried out over some period of time, whereas an alarm query delivers alarm status at some instant in time.

Monitoring Power, S/N

Commands in client-server allow you to monitor the signal-to-noise ratio (SNR), and average power level of a signal arriving on one or more receive timeslots.

Framing Format

GL Server supports commands that allow you to set and/or query the framing format (Unframed, D4 (T1) , ESF(T1), ESF(J1), CAS(E1), FAS(E1), and CRC4) .

```
//Example for FRAME FORMAT SETTINGS is shown below  
set signaling mode cas #*;  
set crc4 on#*;  
set crc4 off #*;  
set unframed on #*;
```

Interface

Commands allow you set and/or query Terminate, Monitor, and Bridge Rx interface options.

```
//Example for INTERFACE SETTINGS  
set rx interface terminate #*;
```

Clock Source

Provides options to set the transmit clock source and query the clock source currently being used by a specified port or ports.

```
//Example for CLOCK SETTINGS is shown below  
set tx clock source internal #*;
```

Loop back

GL Server supports commands that allow user to loop back a DS1 signal at various points.

```
//Example for LOOPBACK SETTINGS is shown below  
set outward driver loopback off #*;  
set inward driver loopback off #*;
```

VF Options

Provides options to set and/or query the VF options such as gain, encoding, and drop and insert VF ports.

GL Server and Client Functions

“**Actions**” are performed immediately in response to the “**Commands**” from the clients. The Server informs the Client on tasks: started, status, complete. Typical tasks include BER testing, File/Digits/Protocol Frames playback and recording, Alarm monitoring, and so on.

Client programs communicate with the GL Server via TCP/IP-encapsulated commands and responses. WCS clients are available via console/terminal (CL) and accessible remotely through SSH or from Python scripts running in Windows® or Linux® operating system. Other environments commonly used for client development include Perl, MatLab, Visual Basic, and LabView.

Windows/Linux Python Client Module

The Windows Client Python Module (WCPM) **wcli_t1_e1.pyd** provides a client interface to the T1 E1 WCS servers. It allows a Python program to connect to one or more GWSs and execute server commands, asynchronous tasks and receive replies from GWSs. WCPM enables connection to multiple GWSs allowing the Python program to control T1 E1 equipment at multiple computers and geographical locations via TCP/IP. Supports Python versions 3.6 or above.

All the complexity of TCP/IP communications is completely hidden from users and is reduced to invoking functions with string parameters and functions returning string responses.

WCPM complies with Python naming convention standards and development practices. When WCPM functions are invoked incorrectly a proper exception is thrown making it easy to diagnose and correct the problem.

The Linux Client Python Module **wcli_t1_e1.so** can be used on any Linux operating system.

Real world applications include:

- Support for framed and unframed transmission/reception of signals
- Intrusive/Non-Intrusive T1 E1 Testing
- Performance monitoring and testing of multiple site locations from a single client
- Shared use of T1 E1 test equipment from multiple client locations
- Automated factory testing on production lines
- Integration of T1 E1 testing into more complex testing systems

User can observe response in Python shell as shown below.

```

Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\Aditya\Desktop\sample.py =====
*****rx Interface terminate*****

Set rx interface terminate:
OK
Get rx interface :
#1.rx_intfc=terminate; #2.rx_intfc=terminate
-----

*****HDB3 line coding*****

Set line coding hdb3:
OK
Get line coding:
#1.line_cd=hdb3; #2.line_cd=hdb3
-----

*****VF Input on/off*****

Set VF input on card#1
OK
Get VF input on card#1
#1.vf_in=on 1
-----

***** Signaling bits *****

go 1,1,1,1 #1:
OK
get Signaling bits on card#2 TS:1..10
#2:1.sig_bits=1,1,1,1; #2:2.sig_bits=1,1,1,1; #2:3.sig_bits=1,1,1,1; #2:4.sig_b
its=1,1,1,1; #2:5.sig_bits=1,1,1,1; #2:6.sig_bits=1,1,1,1; #2:7.sig_bits=1,1,1,1
; #2:8.sig_bits=1,1,1,1; #2:9.sig_bits=1,1,1,1; #2:10.sig_bits=1,1,1,1
-----

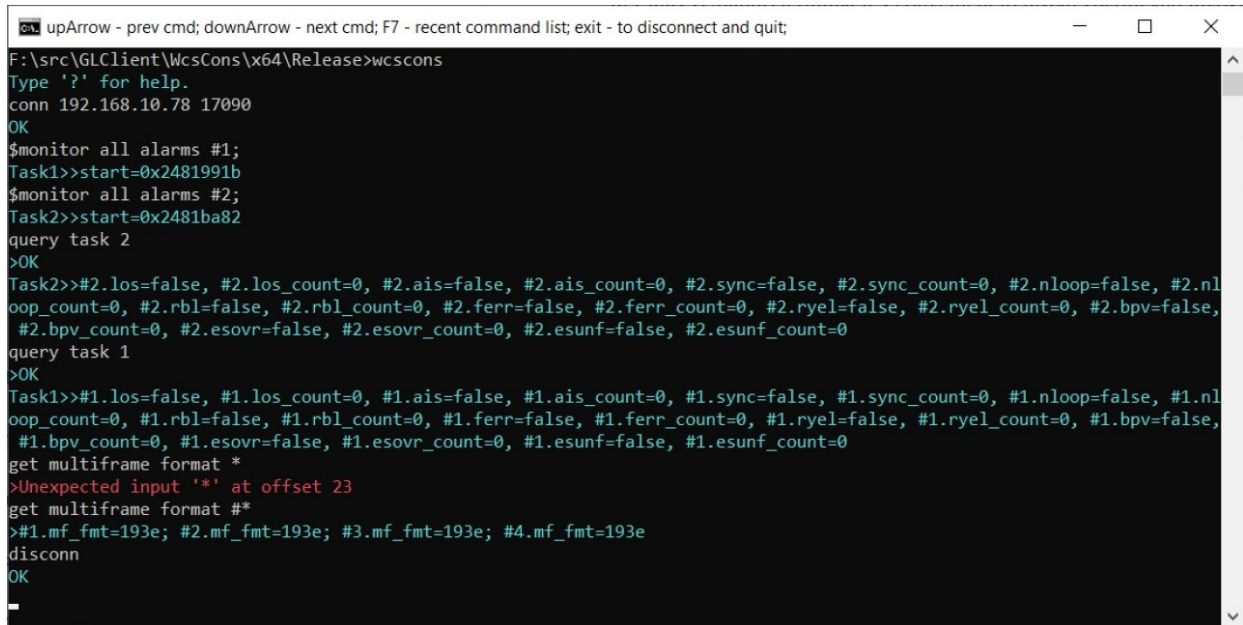
```

Windows/Linux Client

Windows/Linux Client (WLC) is a Command Line Interface (CLI) application that issues commands to T1 E1 WCS servers and display replies into Console/PowerShell/Terminal Windows. WLC works in Windows and Linux versions. However, through SSH or another remote access terminal it can be used on any operating system. WLC is a portable Windows/Linux WCS client communication library compatible with WCS server.

WLC has two groups of commands:

- **WLC Commands:** Display WLC command help, connect to WCS Server, disconnect from WCS Server, pass commands to WCS servers and display replies, save, and execute scripts etc.
- **WCS Commands:** Controls T1 E1 cards and display responses from the WCS server commands and tasks



```

upArrow - prev cmd; downArrow - next cmd; F7 - recent command list; exit - to disconnect and quit;
F:\src\GLClient\WcsCons\x64\Release>wcscons
Type '?' for help.
conn 192.168.10.78 17090
OK
$monitor all alarms #1;
Task1>>start=0x2481991b
$monitor all alarms #2;
Task2>>start=0x2481ba82
query task 2
>OK
Task2>>#2.los=false, #2.los_count=0, #2.ais=false, #2.ais_count=0, #2.sync=false, #2.sync_count=0, #2.nloop=false, #2.nloop_count=0, #2.rbl=false, #2.rbl_count=0, #2.ferr=false, #2.ferr_count=0, #2.ryel=false, #2.ryel_count=0, #2.bpv=false, #2.bpv_count=0, #2.esovr=false, #2.esovr_count=0, #2.esunf=false, #2.esunf_count=0
query task 1
>OK
Task1>>#1.los=false, #1.los_count=0, #1.ais=false, #1.ais_count=0, #1.sync=false, #1.sync_count=0, #1.nloop=false, #1.nloop_count=0, #1.rbl=false, #1.rbl_count=0, #1.ferr=false, #1.ferr_count=0, #1.ryel=false, #1.ryel_count=0, #1.bpv=false, #1.bpv_count=0, #1.esovr=false, #1.esovr_count=0, #1.esunf=false, #1.esunf_count=0
get multiframe format *
>Unexpected input '*' at offset 23
get multiframe format #*
>#1.mf_fmt=193e; #2.mf_fmt=193e; #3.mf_fmt=193e; #4.mf_fmt=193e
disconn
OK

```

Optional Windows Client Server (WCS) Modules

WCS Module XX610 – File based Record/Playback

- Simultaneously perform simultaneous and synchronous transmit and receive tasks on all the 4 ports (124 TS)
- Transmit / receive streams of various bandwidth (hyper channel, timeslot) with continuous / limited capture
- Allows broadcast file into multi timeslots

Sample Script ://Transmit only or Transmit and receive simultaneously on all the timeslots at a time on each of 4 ports
rx server file "rec1.ala" #1 :1..31 1000 ;
tx server file "qrss.ber" #2 :1..31;

WCS Module XX610 – Client Data Tx Rx

It provides the capability for "clients" to Tx Rx T1 E1 data to Server, that is users can Tx Rx files from the Client location (in case that the Server and Client are not on the same PC) directly to the server.

WCS Module XX629 – ISDN Emulation

This is an optional client application that is used to perform various ISDN Emulation functions –

- Configuring the ISDN Layer parameters, Called/Calling Numbering Plan/Type, Simple NFAS setup for T1, and others
- Place call or accept call for each timeslot or for the whole trunk, Switch and Subscriber Emulation

Sample script for ISDN Emulation:
inform task "SetISDNProt USA AT&T#4ESS Subscriber; inform task "SetISDNProt USA AT&T#4ESS Switch #2";
inform task "StartDChan #1..2";// Start D Channel
inform task "PlaceCall 5551234 5551000 #1:1";
inform task "AnswerCall #2:1";

WCS Module XX625 – Channel Associated Signaling Simulation

- This is an optional client application that simulates any user defined CAS protocol by providing signaling bit transitions and forward/backward frequency tones/digits
- Supported protocols include- E1 MFC-R2 (All variants, fully/semi compelled), E1 European Digital CAS (EUC), T1 Winkstart (R1 wink), T1 Loopstart, T1 Groundstart, and user-defined

WCS Module XX620 – Transmit/Detect digits

- Transmit / Detect DTMF/MF/MFR2-f/MFR2-b digits on channels as they occur
- Cards / software have the ability to transmit and detect digits on all the timeslots

Sample script for Transmit/Detect digits:

```
monitor dtmf digits #1,2,3,4,5,6,7,8,9,10:1..31;...
tx dtmf digits ("1 2 3 4 5 6 7 8 9 0 * # a b c d ",-10, 20,20) #1,2:1..31;
tx dtmf digits ("1 2 3 4 5 6 7 8 9 0 * # a b c d ",-10, 10,10) #3,4:1..31;
tx dtmf digits ("1 2 3 4 5 6 7 8 9 0 * # a b c d ",-10, 10,10) #5,6:1..31;
tx dtmf digits ("1 2 3 4 5 6 7 8 9 0 * # a b c d ",-10, 10,10) #7,8:1..31;
tx dtmf digits ("1 2 3 4 5 6 7 8 9 0 * # a b c d ",-10, 10,10) #9,10:1..31;
```

- This is an optional client application that simulates any user defined CAS protocol by providing signaling bit transitions and forward/backward frequency tones/digits
- Supported protocols include- E1 MFC-R2 (All variants, fully/semi compelled),E1 European Digital CAS (EUC), T1 Winkstart (R1 wink), T1 Loopstart, T1 Groundstart, and user-defined

```
E1_Regressiontest.gls - GL Client
File Edit View Connect Script Log User Help
Task 1: #2:1[0.000]: idle, pwr=66.1, dur=3285.0
Task 1: #2:1[3.285]: dtmf-1, pwr=-10.0, f1=698, p1=-13.0, f2=1210, p2=-13.0, dur=39.8
Task 1: #2:1[3.325]: idle, pwr=-56.9, dur=60.3
Task 1: #2:1[3.385]: dtmf-2, pwr=-10.0, f1=698, p1=-13.0, f2=1337, p2=-13.0, dur=39.9
Task 1: #2:1[3.425]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.485]: dtmf-3, pwr=-10.0, f1=698, p1=-13.0, f2=1478, p2=-13.0, dur=40.0
Task 1: #2:1[3.525]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.585]: dtmf-4, pwr=-10.1, f1=771, p1=-13.1, f2=1210, p2=-13.1, dur=40.0
Task 1: #2:1[3.625]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.685]: dtmf-5, pwr=-10.0, f1=771, p1=-13.0, f2=1337, p2=-13.0, dur=40.0
Task 1: #2:1[3.725]: idle, pwr=-53.6, dur=60.3
Task 1: #2:1[3.785]: dtmf-6, pwr=-10.0, f1=771, p1=-13.0, f2=1478, p2=-13.0, dur=39.8
Task 1: #2:1[3.825]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.885]: dtmf-7, pwr=-10.0, f1=853, p1=-13.0, f2=1210, p2=-13.0, dur=39.9
Task 1: #2:1[3.925]: idle, pwr=-66.1, dur=60.1
Task 1: #2:1[3.985]: dtmf-8, pwr=-9.9, f1=853, p1=-13.0, f2=1337, p2=-13.0, dur=40.0
Task 1: #2:1[4.025]: idle, pwr=int, dur=60.0
Task 1: #2:1[4.085]: dtmf-9, pwr=-10.0, f1=853, p1=-13.0, f2=1478, p2=-13.0, dur=40.0
Task 1: #2:1[4.125]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[4.185]: dtmf-0, pwr=-10.0, f1=942, p1=-13.0, f2=1337, p2=-13.0, dur=40.0
Task 1: #2:1[4.225]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[4.285]: dtmf-, pwr=-10.1, f1=942, p1=-13.1, f2=1210, p2=-13.1, dur=40.0
Task 1: #2:1[4.325]: idle, pwr=-53.3, dur=60.0
Task 1: #2:1[4.385]: dtmf-#, pwr=-9.9, f1=942, p1=-12.9, f2=1478, p2=-12.9, dur=40.0
Task 1: #2:1[4.425]: idle, pwr=-66.1, dur=60.0
Task 2: 12800 bytes transmitted
Task 2: Task 2 complete

monitor dtmf digits #2:1;
tx dtmf digits("1 2 3 4 5 6 7 8 9 0 * # A B C D ",-10,40,60)#1:1;
end task *;

Ready Ver 3 B NUM
```

WCS Module XX625 – Channel Associated Signaling Simulation

- This is an optional client application that simulates any user defined CAS protocol by providing signaling bit transitions and forward/backward frequency tones/digits
- Supported protocols include- E1 MFC-R2 (All variants, fully/semi compelled),E1 European Digital CAS (EUC), T1 Winkstart (R1 wink), T1 Loopstart, T1 Groundstart, and user-defined

WCS Module XX012 – Jitter Generation, Measurement and Pulse Mask

The Pulse Mask module can detect, monitor, and plot transmitted pulses using simple commands . Some of the supported commands are CheckPulseMask, validate pulse, and stop pulse mask validation.

The Jitter measurement can be performed through commands. The tasks include monitoring jitter ranges, and getting jitter ranges on a specified port.

Sample Script for pulse detection and validation (E1)
 run task "WcsPulseMaskE1:CheckPulseMask" #2 continuos;
 inform task 1 "validate pulse" ;
 inform task 1 "validate pulse dump";
 wait 8000 msec;
 end task 1;

Sample Script for jitter measurement
 get jitter ranges #*;
 monitor jitter range 8000 hz #1;
 monitor jitter range 8000 hz #1 1 report 1 min;
 monitor jitter range 1000 hz #1 24 report events cfg "JitterMonitor.ini";

WCS Module XX626 - SS1 Signaling Analysis and Emulation

- The SS1 Module generates and analyzes tone sequences that make up SS1 dial digits. Sequences of pulse and guard tones are detected, decoded, and assembled into their corresponding dial digits

WCS Module XX630 - DSP Functionality

Scripted DSP commands provide the ability to specify a sequence of digital signal processing steps to be performed on incoming and/or outgoing timeslots. Visit [Scripted DSP Capability in WCS](#) for more details. List of some sample DSP scripts are as below -

- Transmit filtered tones and white noise
- Adding speech and noise to the receive data
- Double talk simulation for echo canceller testing
- Delay and attenuate the transmitted tone and more
- Dynamic and Static DSP operations

Sample Script transmits sum of white noise , tone , "qrss.ber" file
 tx(dtmf digits ("1234567890*#abcd12345", -10, 50, 60), #2:1..2,broadcast) 15 sec;
 tx(sum(whitenoise(-10),tone(1000,-10),infile("qrss.ber")),#1:1) 10 sec;

WCS Module XX631 – Dynamic DSP Capability

Scripted DSP commands provide the ability to perform dynamic or time-varying operations via schedules. A schedule is a CSV file specifies a sequence of digital signal processing steps to be performed at specified time offset for each operator on incoming and/or outgoing signals. Functions include Amplification/Attenuation, Delay, Filter, File Tx/Rx, Logical operations, Echo Cancel, and so on.

Sample script for Offline Amplification/Attenuation
 outfile(atten(tone(642 hz, -10 dbm), 20 db), "WinClientServer\DynDspOp\DynScaleDspOp.ala") 60 sec cfg
 "WinClientServer\DynDspOp\DynDspOp.ini";

WCS Module XX634,XX640, XX641 for HDLC

Multi-Channel HDLC Emulation and Analysis and File based High Throughput HDLC Record/Playback (XX634)

HDLC Record/Playback (HdlcHpio) module permits:

- Receive / transmit of HDLC streams of various bandwidths (hyper channel, timeslot, and multiple sub-channel streams per timeslot)

HDLC Emulation and Analysis (HdlcTerr) module permits:

- Transmission of memory generated sequences of fixed / variable length HDLC frames, HDL Trace file frames, various bandwidth streams, and introduce various impairments

Sample Script to check for tx & rx operations with HdlcHpio

```
run task "HdlchpioE1:TxRx" #1..2:1;
inform task 1 "Rx 'C:\Ts.hdl' 500000 CONT TS ";
inform task 1 "Tx 'C:\10sec_1flags.hdl' EOF TS FLAGS 1";
inform task 1 "start";
query task 1;
stop task 1;
```

File based HDLC Record/Playback and Remote HDLC capture (XX640, XX641)

This module allows transmission/reception of HDL frames files located on the server and client.

This optional module allows:

- Displays the command syntax
- Transmitting /Receiving HDLC frames in HDL files located on the server / client

Sample Script to generate proper hdlc frames with HdlcTerr

```
run task "HdlcTerrE1:Tx" using "FRAMES 1000 SEQNUM MSB2 FLAGS 100" #1..2:31;
//run task "HdlcTerrE1:Tx" using "HC SEQNUM MSB2 FRAMES 1000 " #1..2:1..2;
//run task "HdlcTerrE1:Tx" using "SEQNUM MSB2 FRAMES 1000 " #1..2:1:1,2;
inform task 1 "start";query task 1;
```

WCS Module XX635, XX636 - PPP, MLPPP, and Multi-Channel (MC) MLPPP Emulation and Analysis

This optional client application can accomplish PPP, MLPPP, and Multi-Channel (MC) Emulation and Analysis . It displays the command syntax, sends and receives MLPPP frames with or without impairments, and logs the events.

Sample Script for MLPPPTerr is as shown below:

```
run task "MLPPPTerrE1:Tx" using " EOF HDLFILE 'c:test.hdl' SHORTSEQNUM";
inform task 1 "TS #1:1.10 MRU 128";
inform task 1 "TS #1:11.20 MRU 215";
inform task 1 "start"; // start transmission
end task 1; // explicitly end the task;
```

WCS Modules XX645, XX646 for TRAU

File based TRAU Record/Playback (XX645)

This WCS module supports TRAU Traffic Capture/Playback, with additional TRAU broadcasting transmission and multiple receive features per task.

Multi-Channel TRAU Tx/Rx Emulation and Analysis (XX646)

This is (TrauTerr) is a high performance, high precision and high flexibility Windows Client Server (WCS) error-testing task for TRAU. The frames are generated based on the codec type, impairments and time alignment (if specified) and transmitted through the 16 or 8 kbps channels. The module allows:

- Displays the command syntax
- Sends TRAU frames with or without impairments, and time alignment
- Receives TRAU frames and logs the errors

Sample Script for receiving TRAU frames into an hdl file

//For T1 Card:

```
run task "TrauFuncT1:RxFile" using "trau.hdl 10000000 CONT 16K SC:FF UPLINK " #1:23;
```

//For E1 Card:

```
run task "TrauFuncE1:RxFile" using "trau.hdl 10000000 CONT 16K SC:FF UPLINK " #1:23;
```

//This will receive 4 streams of 16Kbps each on TS 23 cards 1 continuously.

Sample Script for Trau Tx TRAU frames using 16kbps for GSM6.10

```
run task "TRAUTerrE1:TxRx";
```

```
inform task 1 "SC #1:1:1..2 RATE 16K DIRECTION UPLINK CODEC GSM610";
```

```
inform task 1 "START TX #1:1:1..2 FRAMES 100 SEQNUM MSB4";
```

WCS Module XX665 - File based HDLC Record/Playback over Sa-bits (E1 Maintenance Data Link)

The File based HDLC Record/Playback over SAbits (SaBitsFunc) module is an optional application that allows receive and transmit of HDLC frames in *.hdl file format.

Sample Script for E1 Maintenance Data Link:

//Transmission of first three frames on card 1 with 1000 flags between frames

```
run task "SaBitsFuncE1:SaBitsTxFile" using "'c:\data files\test.hdl' CONT" #1;
```

```
//Continuously transmit file c:\data files\test.hdl'
```

WCS Module XX655 - Multi-link Frame Relay Emulation

Automated testing of FR and MFR can be accomplished using client-server based Multi-link Frame Relay Emulation module. The MFR Emulation module is an optional application that displays the command syntax, sends and receives FR and MFR frames with or without impairments.

Sample Script for creating links is shown below:

```
run task "MFREmulatorE1:TxRx";
```

```
inform task 1 "SIMULATION FR";
```

```
inform task 1 "HC #1:1..31 FLAGS 100";
```

```
inform task 1 "ACTIVATE HC #1:1..31";
```

```
inform task 1 "CREATE VC HC #1:1..31 DLCI 1 FRAG FORMAT END TO END FRAGSIZE 256";
```

```
inform task 1 "Tx: HC #1:1..31 DLCI 1 CONT FIXLEN 1500 SEQNUM MSB4";
```

```
inform task 1 "START TX HC #1:1..31 DLCI 1";
```

```
query task 1;
```

```
end task 1;
```

WCS Module XX654 - Inverse Multiplexing over ATM

Inverse Multiplexing for ATM (IMA) can be emulated on up to 16 T1 or E1 lines using GL's client-server based IMA Emulation software module. All the operations that are performed using GUI can also be performed in CLI making it suitable for automated testing.

Sample Script for sending sequence number traffic:

```
run task "AtmImaServerT1:StartAtmIma";
inform task 1 "ENABLE ICP";
inform task 1 "CONFIG: IMAID 1 IMAFRAMELEN 128 SYMMETRYMODE 0";
inform task 1 "ADDLINK: HC #1:0..23 LINKID 0";
inform task 1 "ADDLINK: HC #2:0..23 LINKID 1";
inform task 1 "START";
inform task 1 "Rx: VPI 100 VCI 200 AALTYPE 5 CONT FIXLEN 1500 SEQNUM LSB4 0 1";
inform task 1 "Tx: VPI 100 VCI 200 AALTYPE 5 CONT FIXLEN 1500 SEQNUM LSB4 0 1";
query task 1;
end task 1;
```

WCS Module XX660 – File based Record/Playback over FDL

The File based Record/Playback over FDL (FdlFunc) module is an optional application that allows to receive and transmit HDLC frames and signals in *.hdl file format over facility data link (FDL).

Sample Script for Transmitting and Receiving HDLC frames and signals over FDL using HDL file:

```
run task "FdlFuncT1:FdlTxFile" using "c:\data files\test.hdl" CONT 4096 " #1 continuous;
//Continuously transmit file c:\data files\test.hdl' on card 1 with 4096 signal repetition for each signal in the HDL file.
run task "FdlFuncT1:FdlRxFile" using "c:\test.hdl" 1000000 100000" #1 continuous;
//receive into file c:\test.hdl (allocate 10MB disk space for the file) up to 100000 frames on card 1.
```

WCS Module XX670 – Multi-Channel BER Testing

Multi-channel BER Testing server function allows the user to take advantage of the more advanced features of the HD boards.

- Supports transmit and receive of *.ber files
- Bit inversion on Rx pattern
- Logic errors insertion for Tx and Rx patterns

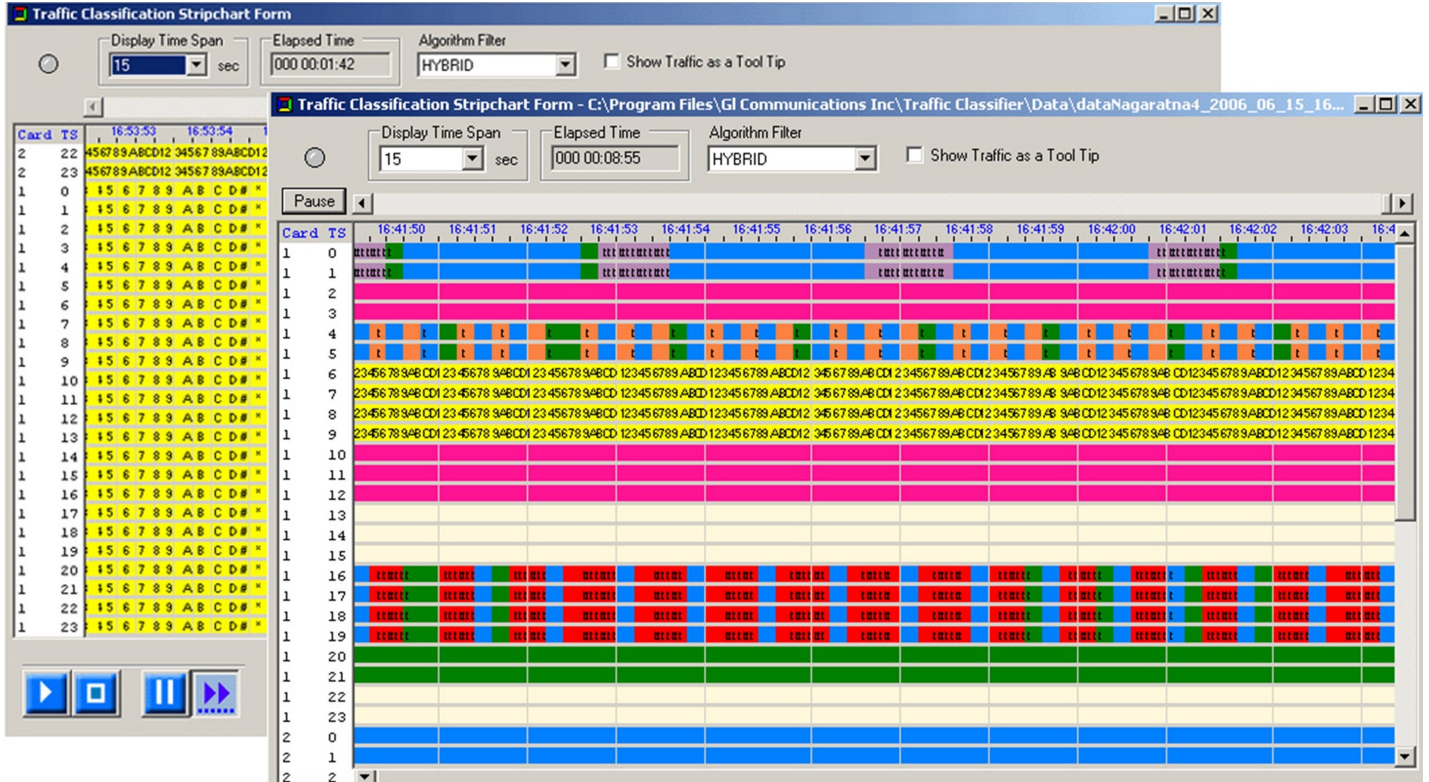
*Sample Script for Multiple channels-single stream N*64Kbps (N=5 in this case)*

```
bert 64k pattern "qrss" #1: 1..5 continuous;
run task "BerFuncE1:RxFile" using " 'c:\qrss.ber' LOG 'c:\qrss.log' HC " #2:1..5;
query task 1;
end task 1;
```

WCS Module XX680 – Traffic Classifier

Traffic Classifier is an application that can analyze the traffics such as voice, fax, data, and tones (dial tone, ring-back tone, busy tone, so on) and also identify dialing digits and other events happening on a T1 E1 network.

- Ability to non-intrusively monitor the traffic on a T1 or E1 line in real-time and offline modes
- Codecs supported - μ -law, A-law, and linear



Fax Emulation – 2 to 120 Fax Ports (XXXFT0)

The software can transmit and receive the fax information as electrical signals over the T1 or E1 lines. The contents (text or images) are sent as a graphic image. The receiving end reconverts the coded image and creates a copy of the document.

Sample Script for Fax Emulation is as shown below:

```
run task "FaxSimulatorE1:StartFaxSim";
```

```
inform task 1 "START";
```

```
inform task 1 "TXFAX #1:1 TIFF_FILE 'WinClientServer\FAX Simulator\send\3.tif' CODEC_TYPE ALAW MODEM_TYPE 1  
MIN_RATE_TYPE 2400 MAX_RATE_TYPE 4800 PAGESIZE_TYPE 1 RESOLUTION_TYPE 16 ECMENABLED 1";
```

WCS Modules for MAPS™

XX649 – Scripted ISUP Simulator (MAPS™ SS7)

MAPS™ SS7 can simulate Service Switching Point (SSP) and ISUP signaling specification as defined by the ITU-T standards. It supports testing network elements, error tracking, regression and conformance testing, load testing, and generation of high volumes of ISUP traffic.

XX647 Scripted ISUP Conformance Testing (MAPS™ SS7 Conformance)

Suitable for conformance tests and functional tests, where test objects can be accurately, reliably and comfortably validated for compliance with ITU-T standard Q.761-764 and Q.784.

XX649 – Scripted MAP Simulator (MAPS™ MAP)

MAPS™ MAP (Mobile Application Part) can simulate various MAP signaling interfaces (C, D, E, F, H, and and Packet-switched interfaces such as Gc, Gr, Gf, Gd) in GSM/UMTS networks as defined by 3GPP standards.

XX648 – Scripted ISDN Simulator (MAPS™ ISDN)

MAPS™ ISDN can simulate ISDN signaling as defined by the ITU-T standards. Currently it is used to perform testing using ISDN protocol messages over T1 or E1, and offers a complete solution for testing, troubleshooting, and maintenance of devices and networks implementing PRI ISDN.

XX651 - Scripted CAS Simulator (MAPS™ CAS)

MAPS™ CAS (Channel Associated Signaling) simulates CAS signaling over TDM (T1 E1) with the ready scripts on a single or on all timeslots. Calls are established, once the signaling information such as the signaling bits, MF, DTMF, or tones are detected. Optional client application simulating all CAS protocols, including wink-start, loop-start, ground start, R1, MFC-R2, etc.

XX696 – MAPS™ CAP

It is a multi-protocol, multi-technology platform used for the emulation of a variety of communication protocols over IP, TDM, and Wireless networks. It can emulate CAP (CAMEL Application Part) supplementary services such as unified messaging, prepaid, toll-free (Freephone), and fraud control. These services are available in TDM based GSM, GPRS, UMTS over TDM and IP networks.

XX656 – MAPS™ INAP Emulator (MAPS™ INAP)

MAPS™ is an advanced protocol simulator/tester for **INAP (Intelligent Network Application Part)** services. The tester supports testing network elements gsmSCF and gsmSSF, error tracking, regression testing, conformance testing, and load testing/call generation. It is able to run pre-defined test scenarios against INAP test objects in a controlled and deterministic manner

XX624 – Scripted FXO FXS Simulator (MAPS™ FXO FXS)

MAPS™ emulates functions of Foreign Exchange Subscriber (FXS) by analog phone lines and Foreign Exchange Office (FXO) by phones using the FXO and FXS ports on a tProbe™. MAPS™ provides a facility to place call/answer incoming call on both FXO and FXS ports.

XX652 – Scripted MLPPP Conformance Testing (MAPS™ MLPPP)

MAPS™ MLPPP Emulator can simulate a complete PPP/MLPPP link (over T1 or E1) between two peers (Router or a Switch), with MLPPP signaling conforming to IETF specifications.

XX692 – Scripted GSM A Simulator (MAPS™ GSM A)

MAPS™ GSM A can simulate BSSMAP and DTAP messages and signaling specification as defined by 3GPP standards. The tester supports testing network elements MSC and BSS, Error tracking, regression testing, conformance testing, load testing/call generation and generation of high volumes of GSM traffic.

XX693 – Scripted GSM Abis Simulator (MAPS™ GSM Abis)

MAPS™ GSM Abis can simulate BTSM messages and signaling specification as defined by 3GPP standards. Supports testing BTS and BSC, error tracking, regression/conformance/load testing, and generation of high volumes of GSM traffic.

XX690 – SS7 Decode Agent

A console client application that monitors SS7 Links, decodes multiple SS7 protocol standards, filters user-specified protocol parameters, builds CDRs, and streams over TCP/IP to remote site.

XX691 – ISDN Decode Agent

A console client application that monitors ISDN Links, decodes multiple ISDN protocol standards, filters user-specified protocol parameters, builds CDRs, and streams over TCP/IP to remote site.

Buyer's Guide

Item No	Product Description
XX600	Basic Client/Server Scripted Control Software (Included with Basic Software)
XX610	File based Record/Playback
XX616	T1 E1 WCS Client Python Module
XXFT0	WCS Fax Simulator
XX620	Transmit/Detect digits
XX625	Channel Associated Signaling (CAS) Simulation
XX626	SS1 Signaling Analysis
XX630	DSP Functionality
XX631	Dynamic DSP Capability
XX629	ISDN Emulation
XX634	Multi-Channel HDLC Emulation and Analysis and File based High Throughput HDLC Record/Playback
XX635	PPP Emulation and Analysis
XX636	MC-MLPPP Emulation and Analysis
XX640	File based HDLC Record/Playback
XX641	File based HDLC Remote Record/Playback
XX643	MTP2 Emulation
XX646	Multi-Channel TRAU Tx/Rx Emulation and Analysis
XX648	Scripted ISDN Simulator (MAPS™ ISDN)
XX650	File based HDLC Record/Playback over SA-bits
XX651	Scripted CAS Simulator (MAPS™ CAS)
XX654	Inverse Multiplexing for ATM Emulation
XX655	MFR Emulation
XX656	MAPS™ INAP (over T1 E1) Emulator



818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878, U.S.A
 (Web) www.gl.com - (V) +1-301-670-4784 (F) +1-301-670-9187 - (E-Mail) info@gl.com

Buyer's Guide (Contd.)

Item No	Product Description
XX660	File based Record/Playback over FDL
XX670	Multi-Channel BER Testing
XX680	Traffic Classifier
XX690	Console based SS7 Protocol Decode Agent
XX691	Console based ISDN Protocol Decode Agent
XX692	Scripted GSM A Interface Simulator (MAPS™ GSM A)
XX693	Scripted GSM Abis Interface Simulator (MAPS™ GSM Abis)

Item No	Related Software
XX019	Tx/Rx File Utility
XX020	Record and Playback of Files
XX051	Synchronous Trunk Record Playback
XX031	Call Capture and Analysis
XX068	Semi-Automated & Scripted Echo Canceller Testing Suite

Item No	Related Hardware
PTE001	tProbe™ T1 E1 Base Unit
FTE001 , ETE001	Quad and Octal T1 E1 Analyzer Boards
XTE001	Dual Express (PCIe) T1 E1 Boards

For more details, refer to [Windows Client/Server for T1 E1 Analysis](#) webpage.

For more details on MAPS™ products, refer to [Message Automation & Protocol Simulation \(MAPS™\)](#) webpage.



818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878, U.S.A
 (Web) www.gl.com - (V) +1-301-670-4784 (F) +1-301-670-9187 - (E-Mail) info@gl.com