INTRODUCTION to MAPS™
MAPS™

MA - Message Automation

+ 

PS - Protocol Simulation
About MAPS™

- MAPS stands for Message Automation and Protocol Simulation
- It is a generic framework for the generation of telecommunications protocol messages and transmission of bearer traffic.
- MAPS™ is built on a proprietary scripting language developed by GL Communications.
- All MAPS™ products come with out-of-the-box scripts that act as fully functional state machines for the relevant protocol.
About MAPS™

- **Scripts**: Scripts act as the state-machine, or engine for a given call. The logic of what messages to send when is all contained in a script.

- **Messages**: MAPS has an inventory of generic Message Templates (ex: Invite.txt) which it loads from the hard drive when transmitting an actual message. Messages are completely customizable.

- **Profiles**: Scripts and Messages are kept as generic as possible. Specific information (ex: Contact = 12345@sip.carrier.com) about a call is sourced from .xml profiles.
Basic Requirements for Simulation

• **Message Templates**

The message templates are nothing but structure of message stored in particular file format. e.g.: SS7 Protocol suite message template will have “. HDL” format.

• **A ‘Script’**

To send and receive these messages between two nodes and take appropriate actions for a particular message.

• **An ‘Import’ mechanism**

A mechanism for reading the contents of the message template, and replacing the Key Identifier with the value given by the user (or some other means) at the run time. This process of inserting the user values into the message template before sending is called “Import”.


Basic Requirements for Simulation...

- An ‘Export’ mechanism
  A mechanism to extracting Key Identifier values from the received response and store for the future use (in the same call scenario) is called “Export” (This exported value can also be imported to message template in future).

- A ‘Profile’ file
  Once the Key Identifiers are identified for all the message templates in a call scenario, required values are configured for these Key Identifiers in a file called Profile.
Supported Protocols / Interfaces

CAS, ISDN, MLPPP, SS7, IUP, MAP, CAMEL, INAP

SIP, Megaco, MGCP, SIP-I, SIGTRAN (ISUP, ISDN), SKINNY

4G Access (E-RAN)

LTE (X2, S1, eGTP, SGs, SLs)
IMS, Diameter

IMS

3G Access (UTRAN)

UMTS (IuH, IuCS, IuPS, GnGp) GPRS

PBX

tdm / PSTN Network

VoIP (Packet Network)

GL

GL

Signalling and Traffic Generation and Analysis

GL

GL

GL

GL

MAPS™ G – Software-based, or Hardware-based
MAPS™ HD 4G, 10G, 1G – Rackmount Platforms

http://www.gl.com/maps.html
Common Protocol Emulation Framework

LTE Simulation

SS7 Simulation

SIP Simulation
MAPS™ Working Principle

Message Automation and Protocol Simulation (MAPS)

- **Message Templates**
  - Create Call Sequence
- **Scripts**
  - Load Script
- **Profiles**
  - Assign values to variables
  - Load Profile
- **Event Profiles**
  - Load Event Profile

**Call Control**

CALL GENERATION
(Load Scripts, Profiles)

CALL RECEPTION

IP/TDM Network

D U T
Customize Test Scenarios using Scripts

- Unlimited access in creating test scenarios
- Build valid or invalid & conformance test cases.
- A simple, easy to learn but very powerful scripting language.
- Can be an Extremely simple scripts to test a particular scenario. But Flexible enough to simulate a complete protocol state machine.
- A GUI based 'Script Editor' helps to build scripts even before syntax and semantics of the scripting language is familiar.
Sample Script

Send “Initial Address” “InitialAddressImport”;

Recv “Address Complete” “AddressCompleteExport”;

Recv “Answer” “AnswerImport”;

TxRx:tx_TDM file: filename = “Vijay.pcm”;

Send “Release” “ReleaseImport”;

Recv “Release Complete” “ReleaseCompleteExport”;
Customize Protocol Messages

Message Editor

Screen showing protocol messages editor with various parameters and options.
Fine Control over Call Behavior
User Events

Control moves to "Hold" section, after selecting the "Hold" User Event

ActiveUserEvent: Add: "Retrieve";
"Retrieve" User Event is added

Control moves to "Retrieve" section, after selecting the "Retrieve" User Event
Script Content View

```c
// Script Contents

"PlaceCall"(opc,dpc,cic):
    StartChildScript (ISUPScriptId,"ISUP","ISUP.gls",LoadedProfileName);
    ActiveUserEvent:Add: "Terminate Call","Initiate Reset","Clear Call";
    Status = "ISUP Call Initiated";
    ISUPState = "ISUP CALL INITIATED";
    (ISUPScriptId)goto "ISUPMakeCall":cic,opc,dpc,SLS,NetInd,ConnectionId,StreamID,
    return;

"OnISUPCallInitiated"(opc,dpc,cic):
    ReportEvent (InitialAddress = "Initial Address");
    resume;

"OnISUPCallProgressReceived";
    ReportEvent (AddressComplete = "Address Complete");
    resume;

"OnISUPCallConnected":
    Result="Pass";
    ReportEvent (Answer = "Answer");
    Status = "ISUP Call Connected"
    ISUPState = "ISUP CALL CONNECTED";
    if (StopAll==1)
        goto "Terminate Call":Cause;
    endif
    ActiveUserEvent:Add:"Hold","Suspend";
    ActiveUserEvent:Remove:"Accept Call","Reject Call";
    if (CallDuration != 0)
        starttimer: CallDuration Timer CallDuration msec;
    else
        goto "Terminate Call":Cause;
```

<
<table>
<thead>
<tr>
<th>Line No</th>
<th>Script Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>goto &quot;PlaceCall&quot;: opc, dpc, clic;</td>
</tr>
<tr>
<td>96</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>95</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>95</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>95</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>95</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>95</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>95</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>6</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>7</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>8</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>9</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>10</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>11</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>12</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>13</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>14</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>15</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>16</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>17</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>18</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>19</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>21</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>95</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>97</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>98</td>
<td>イスUP_Call.gls</td>
</tr>
<tr>
<td>99</td>
<td>イスUP_Call.gls</td>
</tr>
</tbody>
</table>

18
Incoming Call Handler
Incoming Call Handler

MAPS™ searching for appropriate script to load against the received msg

IAM  AnswerCall.gls
ACM
ANM

IAM  CIC=1
Answer call Script is bind to the recv msg with CIC=1

IAM  CIC=1
Another IAM msg is received with CIC=1

IAM  CIC=2
Another IAM msg is received with CIC=2

Answer Call Script ID 1
CIC=1

Answer Call Script ID 2
CIC=2

Recv msg is bind to the same script

Binds to the new script with ID=2

Script Loaded
Load Generation

• Stability/Stress and Performance testing using Load Generation
• Different types of Load patterns to distribute load
• User can load multiple patterns for selected script
• User configurable Test Duration, CPS, Maximum and Minimum Call Rate etc
Success Call Ratio Statistics

Call Graph

Call Stats
**Message Statistics**

- **Message Stats** provides a running tabular log of all messages transmitted, retransmitted and received during the session.
- Provides an easy way to monitor the reception of error responses during load generation.
User Defined Graphs and Statistics
Scheduler

MAPS™ Features

Scheduler - default

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Config</th>
<th>Process</th>
<th>Start Time</th>
<th>Duration (Hr:Min)</th>
<th>Emulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>master config</td>
<td>Daily</td>
<td>15:23</td>
<td>1:0</td>
<td>Load Generation</td>
</tr>
<tr>
<td>2</td>
<td>config1</td>
<td>Daily</td>
<td>15:23</td>
<td>1:0</td>
<td>Manual Call Generation</td>
</tr>
<tr>
<td>3</td>
<td>config2</td>
<td>Daily</td>
<td>15:24</td>
<td>1:0</td>
<td>Manual Call Generation</td>
</tr>
<tr>
<td>4</td>
<td>master config</td>
<td>Daily</td>
<td>15:26</td>
<td>1:0</td>
<td>Load Generation</td>
</tr>
<tr>
<td>5</td>
<td>test config</td>
<td>Daily</td>
<td>15:26</td>
<td>1:0</td>
<td>Manual Call Reception</td>
</tr>
</tbody>
</table>

Duration:
- Hours: 13, Minutes: 26
- Hours: 1, Minutes: 0
Traffic Simulation

(for details see http://www.gl.com/traffic-simulation.html)

Scales up to 100K to 200K Sustained Calls

(w 4 x 1 GigE NIC Cards)

Maps™ High Density (HD) Call Generator

“All Interfaces”

“All Protocols”

“Large Capacities”

HD Voice

GPRS, S1, eGTP, X2, SGs, Diameter

Gb, UMTS, luCS, lppS, luH, GnGp

SIP, SIP-I, Megaco, MGCP, SIGTRAN (SS7, ISDN)

CAS, ISDN, SS7, IUP, MLPPP

GSM, Abis, BICC, MAP, CAP, INAP

Remote Scripting and Client Access

GTP Traffic

RTP Media

IP Networks

TDM, PSTN Networks

2G, 3G, 4G Networks

Remote Scripting and Client Access

- Voice Files (wav, pcm, GL’s Proprietary GLW)
- Video Trace Files (hdl - GL’s Proprietary format)
- DTMF and MF Digits
- Tones (single, dual and user-defined)
- Industry Standard Codes (a-law, u-law, gsm and more)
- FAX (T.30 RTP pass-through and T.38 UDPTL )
- Impairments (Latency, Packet Loss, Packet Effects)
- Mobile Data, Packet, GTP Gateway Traffic
Supported RTP Codecs

(see [http://www.gl.com/voice-codecs.html](http://www.gl.com/voice-codecs.html) for details)

- **PCMU/PCMA**: 64kbps, 8000Hz, VAD
- **G.722/G.722.1**: 24/32/64kbps, 16000Hz, No VAD
- **G.729/G.729B**: 8kbps, 8000Hz, VAD
- **GSM 6.10 FR**: 13.2kbps, 8000Hz, No VAD
- **GSM EFR**: 12.2kbps, 8000Hz Yes VAD
- **GSM**: 5.6kbps, 8000Hz, Yes VAD
- **G.726**: 16/24/32/40kbps, 8000Hz, Yes VAD
- **AMR**: 4.75/5.15/5.9/6.7/7.4/7.95/10.2/12.2kbps, 8000Hz, Yes VAD (*OPTIONAL LICENSE*)
- **AMR WB**: 4.75/5.15/5.9/6.7/7.4/7.95/10.2/12.2kbps, 16000Hz, Yes VAD (*OPTIONAL LICENSE*)
- **EVRC**: 1/8, 1/2, 1 rate, 8000Hz, No VAD (*OPTIONAL LICENSE*)
- **EVRC_B**: 1/8, 1/2, 1 rate, 8000Hz, Yes VAD (*OPTIONAL LICENSE*)
- **EVRC_C**: 1/8, 1/2, 1 rate, 16000Hz, Yes VAD (*OPTIONAL LICENSE*)
- **SMV**: Modes 0, 1, 2 and 3, 8000Hz, No VAD (*OPTIONAL LICENSE*)
- **ILBC**: 15.2/13.33kbps, 8000Hz, No VAD
- **SPEEX**: 8kbps, 8000Hz, Yes VAD
- **SPEEX WB**: 11.2kbps, 16000Hz, Yes VAD
TDM Traffic Simulation

Tx
- Pre recorded PCM files
- DTMF, MF, MFR2B and MFR2F Digits
- User Defined Tones
- FAX
- AAL2

Rx
- PCM files
- DTMF, MF, MFR2B and MFR2F Digits
- User Defined Tones
- FAX
- AAL2
RTP Traffic Simulation

IP Network

MAPSTM

Voice Channels

PSTN/IP Phones

Tx
- Pre recorded GLW files
- DTMF, MF Digits
- User Defined Tones
- Insert Voice
- FAX T.30

Rx
- GLW files
- DTMF, MF Digits
- User Defined Tones
- FAX T.30
Single Interface Simulation
Multi Interface Simulation

SSP --> SS7 Point Code

SS7

SS7 Point Code --> Signaling Gateway

SIGTRAN

Signaling Gateway --> Media Gateway

MEGACO

Media Gateway --> MAPS Multi Interface Gateway

RTP

MAPS Multi Interface Gateway --> IP Phone

IP: 192.168.1.X
Multiple Transport Support

- OC-3/STM-1 OC-12/STM-4 Card
- Quad/Octal T1 E1 PCIe Card
- Dual T1 E1 Express (PCIe) T1/E1 Card
- tProbe™ T1/E1/VF/Datacom/ FXO/FXS Analyzer
- USB Dual T1 E1 Analyzer
- USB based Dual T3 E3 Analyzer Unit
IP Hardware

- IP variants of MAPS™ can be run on any modern Windows server.
- A typical i7 platform will be able to handle ~2000 concurrent RTP sessions through a conventional server-grade NIC.
- We also offer an HD (High Density) appliance which can deliver up to 20,000 concurrent RTP sessions per U of rack space.
High Density (HD) RTP Traffic Simulation

- Rackmount network appliance with 4x1GigE NIC
- Transport over UDP and TCP, IPv4 and IPv6, and TLS for secure transport
- Easily achieve up to 20,000 endpoints per appliance (5000 per port)
- Up to 350 calls per second (with RTP traffic)
- Scales to around 100,000 to 200,000 endpoints with use of Master Controller for single point of control
- Manage 10+ MAPS™ systems with single point of control from Master Controller
Introduction to MAPS™ Configurations

- Testbed Setup
- Global Configuration
- Profiles

MAPS™ SS7

Subscriber

SSP

GL

Point Code

CONNECTIONS

MAPS™ SS7

Subscriber

SSP

GL

Point Code

LINK SET
L1 - T1E1 Port, TimeSlot, SLC.
L2 - T1E1 Port, TimeSlot, SLC........

CIC Mapping
CIC -- Port TS
...
...

DUT
Testbed Configuration
Sample Global Configuration

<table>
<thead>
<tr>
<th>Config</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Call Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Call Answer Time in msec</td>
<td>1000</td>
</tr>
<tr>
<td>Call Duration in msec</td>
<td>30000</td>
</tr>
<tr>
<td>Inter Call Duration in msec</td>
<td>1000</td>
</tr>
<tr>
<td><strong>Randomization Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Enable Randomization in msec</td>
<td>Disable</td>
</tr>
<tr>
<td>Minimum Call Duration in msec</td>
<td>1000</td>
</tr>
<tr>
<td>Maximum Call Duration in msec</td>
<td>60000</td>
</tr>
<tr>
<td>Minimum Answer Call Duration in msec</td>
<td>1000</td>
</tr>
<tr>
<td>Maximum Answer Call Duration in msec</td>
<td>30000</td>
</tr>
<tr>
<td>Minimum Inter Call Duration in msec</td>
<td>5000</td>
</tr>
<tr>
<td>Maximum Inter Call Duration in msec</td>
<td>20000</td>
</tr>
<tr>
<td>Enable Location Update</td>
<td>Enable</td>
</tr>
<tr>
<td><strong>Enable or Disable Failure Procedures</strong></td>
<td></td>
</tr>
<tr>
<td>Enable Call Control Failure Parameters</td>
<td>False</td>
</tr>
<tr>
<td>Call Control Failure Cause</td>
<td>User alerting, no answer</td>
</tr>
<tr>
<td><strong>MSHA Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Dynamic Registration</td>
<td>Not Required</td>
</tr>
<tr>
<td>Destination Audit</td>
<td>Not Required</td>
</tr>
<tr>
<td><strong>MSHA Specific Timers</strong></td>
<td></td>
</tr>
<tr>
<td>TaskTimer in msec</td>
<td>2000</td>
</tr>
<tr>
<td>TDBTimer in msec</td>
<td>30000</td>
</tr>
<tr>
<td>DAULTimer in msec</td>
<td>5020</td>
</tr>
<tr>
<td><strong>SCMG Timers</strong></td>
<td></td>
</tr>
<tr>
<td>LltaUst in msec</td>
<td>30000</td>
</tr>
<tr>
<td><strong>IuCS Protocol Specific Timers</strong></td>
<td></td>
</tr>
<tr>
<td>Tribs in msec</td>
<td>10000</td>
</tr>
<tr>
<td>T309 in msec</td>
<td>30000</td>
</tr>
<tr>
<td>T319 in msec</td>
<td>30000</td>
</tr>
</tbody>
</table>
Sample Profile
Local and Global Variables

Test Bed

Global Profiles

Global Variables

Profile 1

List of Instructions

Local Variables

Profile n

List of Instructions

Local Variables
MAPS™ Scripting
MAPS™ Scripts can be written in different ways as we have flexible commands such as **Go to**, **IF Else IF**, **Timers**, **Actions**, **User Events** etc.

**Two Types of Scripting**
- **Simple, Non Event driven**
- **Event Driven**

**Non Event driven:** Defines flow sequentially without monitoring any events. These can be small and simple scripts using send and receive actions.

**Event Driven:** Defines flow on basis of user selected events. Using Event Driven scripting one can achieve Protocol State Machines as per protocol specifications.
//Script Description ....................

//Initialization Section
---------
---------

//Action section
Send "MessageName" "ImportFile Name";
Recv "Message" "ExportFile Name";
Result = "Pass";
State = "................";
Status = "................";
Exit;
Script Events

**Message Handler:** On Receipt of any Message Event control move to defined section in script

**Traffic Event:** On detection of any traffic actions, Control move to detected Traffic Event Section like “Digits Detected”, Tone Detected”, etc.

**Timer:** On Expiry of Timer, control moves to respective Timer Expiry section

**User Events:**
- Within scripts: Goto “Label”
- User Intervention: User Event
- Intervention from another Script: Apply Event to another script
Structure of Event Driven Script

// Script Description

// Initialization Section
Initialize Variables

Initialize Message Handler

Message Handler Section
“Message 1”:

“Message 2”:

“Timer Expiry Section”:

“Traffic Handler Section”:

“User Events Section”:
Scripts

Now let’s see how can we achieve the below call flow using MAPS™ scripting.

Send “Initial Address” “InitialAddressImport”;
Recv “Address Complete” “AddressCompleteExport”;
Recv “Answer” “AnswerImport”;
TxRx:tx _TDM file: filename = “Vijay.pcm”;
Send “Release” “ReleaseImport”;
Recv “Release Complete” “ReleaseCompleteExport”;

[Diagram showing call flow with messages: Initial Address Message (IAM), Address Complete Message (ACM), Answer Message (ANM), Release (REL), Release Complete (RLC).]
Sample Script
Understanding Send and Receive Messages
Basic Send Command

Protocol Engine

- **DECODER**
- **IMPORT**
- **ENCODER**

**Message Template**

```
05 12 50 02 32 01 00
01 02 20 01 0A 00 02
08 06 83 10 55 15 11
01 0A 06 81 11 55 15
```

**Called Address Signal** = 8978675400
**Calling Address Signal** = 9987095827

**Message Sent without Impairment**

**DUJT**
Basic Receive/Decode Command

Decode Message

List of Protocol Fields in a Message

- Protocol
- Message Type
- Mandatory Fixed Parameters
- Mandatory Var Parameters
  - Calling Party Number
  - Numbering Plan Indicator
  - Internal Network Number Indic
  - Called Address Signal

Extracted Variables

- Called Address Signal: 5551234
- Called Address Signal: 5551111

Protocol Engine

incoming Message

54 68 47 66 54
0A 00 02 32 43
06 05 03 83 15
10 10 A0 68 10

DECODER

EXPORT

Called Address Signal
Calling Address Signal
Send Action

Import Action

**Scripts**
send "InitialAddress" "InitialAddressImport";

**Message Template - IAM (InitialAddress.HDL)**

```
05 12 50 02 32 01 00
01 02 20 01 0A 00 02 08 06
83 10 55 15 11 01 0A 06 81
11 55 15 32 04 00 .........................
```

**Initial Address Import.txt**

```
ISUP-->Circuit Identification Code = CIC
--------------
ISUP-->MandatoryVarParms-->Called Party Number--> Called Address Signal = CalledNum
--------------
ISUP-->OptionalVarParms-->Calling Party Number--> Calling Address Signal = CallingNum
--------------
```

**SEND COMMAND**

DECODER

IMPORT

ENCODER

**Initial Address**
Scanning Messages…

"Init":
  "ISUPMsgHandler": wait;

"ISUPMsgHandler":
  "MessageType": "DefaultMsgHandler";
  resume;

"Initial Address":
  "AddressComplete": decode "InitialAddressExport";

"Answer":
"Release":
"Release Complete":
"DefaultMsgHandler":

---

**DATA BUFFER**

CIC = 
CalledNumber = 
CallingNumber =

"Init":
  MsgHandler : "ISUPMsgHandler";
  wait;

"ISUPMsgHandler":
  goto Message_Type="DefaultMsgHandler";
  resume;

"Initial Address":
  "Initial Address":
  "AddressComplete":

"Answer":
"Release":
"Release Complete":
"DefaultMsgHandler":

---

**Message Templates**

---

**HDL**

---------------------------------------------------------------------------------------------------
0000 Service Indicator = ....0011 SCCP
0001 DPC = 2.2.2(00010010 ..010000)
0002 OPC = 1.1.1(01...... 00000010 ....0010)
---

**HDL**

---------------------------------------------------------------------------------------------------
0005 Circuit Identification Code = 00000101 ....00 (5)
0007 Message Type = 00000001 Initial address
Mandatory Variable Parameters = ----------
0012 Called Address Signal = 9880098800
Optional Variable Parameters = ----------
0014 Calling Party Number = optional parameter
0012 Calling Address Signal = 9440094400
001E End of optional parameters octet = x00

---
Send Command With Impairment

Protocol Engine

DECODER

Message Template

05 12 50 02 32 01 00
01 02 20 01 0A 00 02
08 06 83 10 55 15 11
01 0A 06 81 11 55 15

IMPARE

Message Template

05 12 50 02 32 01 00
01 02 20 01 0A 00 02
08 06 83 10 55 15 11
01 0A 06 81 11 55 15

ENCODER

Called Ackness Signal = 8978675400
Calling Address Signal = 9987095827

In Message Editor "8th byte with Offset’’ = represents the Message Type Field.

Create MsgTypeImpair.txt file

IMPAIR REP 1 OFFS 7 AND 00
CLI/APIs for Remote Control and Test Automation
CLI for Remote Call Control & 3rd Party Integration

- TCL
- Python
- Java
- VB Scripts

MAPS Client

Network
(Analog, & IP Networks)

TCP/IP

MAPS Server 1

GL Server

MAPS Server n

MAPS IP Server 1

MAPS IP Server n

load mapsclientfc.dll
maps connect 1 192.168.10.32 10024
maps cmd 1 {start "TestBedDefault.xml"}
maps cmd 1 startscript 1 {"Script.gis" "Profile.xml"}
1 # ts =1, cardno = 2
CLI Working Principle

**Client User** (TCL, Python, VBScript, Java, .Net)

- Start Script1
- User Events
- Wait for Response
- Stop Script1

**MAPS Client IFC**

- Storage Space for Script ID
- Command Processor
- Response Processor

**MAPS CLI Server**

- Server Command
- Server Response
Sample TCL Script (Place_Call.tcl) for MAPS™ ISDN
Sample SIP Registration VB Script

```vbnet
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\regasm /codebase C:\Users\VBScript\MapsClientCSAPI.dll
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.33440
Copyright (C) Microsoft Corporation. All rights reserved.

Types registered successfully
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\cd C:\Users\VBScript
C:\Users\VBScript>cscript VBScriptSIPRegistration.vbs
Copyright (C) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

Connecting to Maps CLI Server...
Connection Established

Starting TestBed...
TestBed Started

Loading Profile...
Profile Loaded

Starting Scripts...
Script Started

Registration...
Script stopped
TestBed stopped
Disconnected from the Server
C:\Users\VBScript>.
```
Java Client
Remote MAPS™ Server

- Multi-node and multi-interface simulation from a single GUI
- Suitable for testing any core network, access network, and inter-operability functions
- Single Licensing Server controlling server and client licenses (no. of users)
- Unlimited number of remote client user can be defined at the server
- Admin privileges to control Testbed and access to configuration files for each remote client user
- Remote Client users has privileges to perform all other functions - call simulation, edit scripts/profiles, and view statistics
- Option to license multiple clients either at Remote client systems (MAPS™ Remote Client to control one or more MAPS™ Server - PKS111) or at the MAPS™ Server systems (MAPS™ Server with Multi-user capability - PKS113).
- Simultaneous traffic generation/reception at 100% on all servers
Send Reports to Database

- MAPS™ generated reports can be sent to Database using built in commands
- This helps to monitor and Analyze test Remotely