Introduction to 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
Common Protocol Emulation Framework

LTE Simulation

SS7 Simulation

SIP Simulation
MAPS™ Architecture
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
Call Generation

MAPS™ Features

Loading Scripts and Profiles

Message Sequence

Active Calls
Call Status
Call Events

Decode Message
Fine Control over Call Behavior

Control moves to "Retrieve" section, after selecting the "Retrieve" User Event
User Events

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

```
"Hold":
CallHoldInitiated = 1;
(ISUScriptId) goto "Hold";
resume;
```

Control moves to "Retrieve" section, after selecting the "Retrieve" User Event.

```
"Retrieve":
CallHoldInitiated = 0;
(ISUScriptId) goto "Retrieve";
resume;
```

```
"Suspend":
SuspendInitiated = 1;
(ISUScriptId) goto "Suspend Call";
resume;
```
Script Contents

"PlaceCall"(opc,dpc,cic):
    StartChildScript(ISUPScriptId,"ISUP","ISUP.qls",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 CallDurationTimer CallDuration msec;
    else
        goto "Terminate Call":Cause;
# Incoming Call Handler

[Image of the Incoming Call Handlers Configuration - default window with message names and script names listed, along with a list of scripts: Isup_Call.gls, Isup_Call - Reject.gls, Isup_Call - Forward.gls, Isup_Call - Conference.gls.]

<table>
<thead>
<tr>
<th>Message Name</th>
<th>Script Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signalling Link Test Message</td>
<td>SLTM.gls</td>
</tr>
<tr>
<td>Initial Address</td>
<td>Isup_Call.gls...</td>
</tr>
<tr>
<td>Release</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Reset Circuit</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Continuity Check Request</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Blocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Unblocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Circuit Group Reset</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Circuit Group Blocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Circuit Group Unblocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Release Complete</td>
<td>Rx_CIC_Management.gls</td>
</tr>
</tbody>
</table>

Scripts:
- Isup_Call.gls
- Isup_Call - Reject.gls
- Isup_Call - Forward.gls
- Isup_Call - Conference.gls

Options: Sequence, Random

Buttons: Add, Delete, Clear

[Up, Down arrows]
Incoming Call Handler

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

IAM AnswerCall.gls
ACM
ANM

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=1

2Another IAM msg is received with CIC=

IAM CIC=2

Recv msg is bind to the same script

Answer Call Script ID 1 CIC=1

Answer Call Script ID 2 CIC=2

Binds to the new script with ID=2

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

IAM CIC=1

Another IAM msg is received with CIC=1

IAM CIC=1

2Another IAM msg is received with CIC=

IAM CIC=2

Script Loaded
Call Reception...

Active Calls

Completed Calls

Message Sequence

Decoded Message Details

---

**Message Sequence**

**Active Calls**

**Completed Calls**

**Decoded Message Details**
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

MAPS™ Features

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
Traffic Simulation

(for details see [http://www.gl.com/traffic-simulation.html](http://www.gl.com/traffic-simulation.html))
Supported RTP Codecs

- **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

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

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
Multiple Transport Support

- Dual T1 E1 Express (PCIe) Board
- Quad / Octal T1 E1 PCIe Card
- Dual HD Universal T1 E1 PCI Card
- 16-Port T1 E1 Breakout-Box
- tProbe™ - Portable USB based T1 E1 VF FXO FXS and Serial Datacom Analyzer
- Portable USB based Dual T1 E1 Analyzer Unit
- OC3 / OC12 Channelized and Unchannelized PCIe Card LightSpeed1000™
- Dual Optical Unchannelized (OC3/STM-1 OC12/STM-4)
- Dual GE Channelized
- Portable USB based Dual T3 E3 Analyzer Unit (Channelized and Unchannelized T3 E3)
- Rackmount T3 E3 Analyzer (Channelized and Unchannelized T3 E3)
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 Unit 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
Local and Global Variables

Test Bed

Global Profiles

Global Variables

Profile 1

List of Instructions

Local Variables

Profile n

List of Instructions

Local Variables
Testbed Configuration
Sample Global Configuration

<table>
<thead>
<tr>
<th>Config</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Configuration</td>
<td></td>
</tr>
<tr>
<td>Call Parameters</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>Randomization Parameters</td>
<td></td>
</tr>
<tr>
<td>- Enable Randomization in msec</td>
<td>Disable</td>
</tr>
<tr>
<td>- Minimum CallDuration in msec</td>
<td>1000</td>
</tr>
<tr>
<td>- Maximum CallDuration in msec</td>
<td>60000</td>
</tr>
<tr>
<td>- Minimum AnswerCallDuration in msec</td>
<td>1000</td>
</tr>
<tr>
<td>- Maximum AnswerCallDuration in msec</td>
<td>30000</td>
</tr>
<tr>
<td>- Minimum InterCallDuration in msec</td>
<td>3000</td>
</tr>
<tr>
<td>- Maximum InterCallDuration in msec</td>
<td>20000</td>
</tr>
<tr>
<td>Enable Location Update</td>
<td>Enable</td>
</tr>
<tr>
<td>Enable or Disable Failure Procedures</td>
<td></td>
</tr>
<tr>
<td>Call Control Failure Parameters</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>MMUA Parameters</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>MMUA Specific Timers</td>
<td></td>
</tr>
<tr>
<td>- TaskTimer in msec</td>
<td>2000</td>
</tr>
<tr>
<td>- ThreatTimer in msec</td>
<td>10000</td>
</tr>
<tr>
<td>- DAUDtimer in msec</td>
<td>5000</td>
</tr>
<tr>
<td>SCIMG Timers</td>
<td></td>
</tr>
<tr>
<td>- Tittainf in msec</td>
<td>30000</td>
</tr>
<tr>
<td>IuCS Protocol Specific Timers</td>
<td></td>
</tr>
<tr>
<td>- Tiibi in msec</td>
<td>10000</td>
</tr>
<tr>
<td>- T303 in msec</td>
<td>30000</td>
</tr>
<tr>
<td>- T305 in msec</td>
<td>10000</td>
</tr>
</tbody>
</table>
MAPS™ Scripting
Script Variants

• 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":
---------
---------
Below call flow scenario using MAPS 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”;

---

**Scripts**

<table>
<thead>
<tr>
<th>SSP</th>
<th>SG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Address Message (IAM)</td>
<td></td>
</tr>
<tr>
<td>Address Complete Message (ACM)</td>
<td></td>
</tr>
<tr>
<td>Answer Message (ANM)</td>
<td></td>
</tr>
<tr>
<td>Voice</td>
<td></td>
</tr>
<tr>
<td>Release (REL)</td>
<td></td>
</tr>
<tr>
<td>Release Complete (RLC)</td>
<td></td>
</tr>
</tbody>
</table>
Sample Script

```
// Initialize Variables
ReportEvent(ISUPStatus = "Started");
KeyIdentification option: digits: <option>;
CalDuration = CalDuration;
CalForCurrentTime = CalForCurrentTime;
CallSetupTime = CalSetupTime;
InterCallDurationTimeOut = InterCallDuration;
ISUP.SIP = "ISUP";
Protocol.Standard = "ISUP";

LocalDCState = "LOCAL";
RemoteDCState = "REMOTE";
TDMSessionState = "NOT STARTED";
State = "TOE";
ISUPResult = "Unknown";
Cause = 16;

if (iEnableRandomization == 1)
    InitiateRandomIf(RandomDuration, RandomValue, <RandomMax, <Uniform>>);
    GenerateRandomIf(RandomDuration, CallDuration);
    EventLog (Call Duration = <RandomDuration>, CallDuration);
    InitiateRandomIf(RandomDuration, RandomValue, <RandomMax, <Uniform>>);
    GenerateRandomIf(RandomDuration, InterCallDurationTimeOut);
    EventLog (Inter Call Duration = <RandomDuration>, InterCallDurationTimeOut);
    InitiateRandomIf(RandomDuration, RandomValue, <RandomMax, <Uniform>>);
    GenerateRandomIf(RandomDuration, InterCallDurationTimeOut);
```

GL Communications
Understanding Send and Receive Messages
Basic Send Command
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 Indicator
  - Called Address Signal= 5551234
- Called Party Number
  - Numbering Plan Indicator
  - Internal Network Number Indicator
  - Called Address Signal= 5551111
- Optional Var Parameters

Extracted Variables
- Called Address Signal= 5551234
- Called Address Signal= 5551111

Protocol Engine

DECODER

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

EXPORT

Called Address Signal
- 5551234
- 5551111

Calling Address Signal
DATA BUFFER

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

Send Action

Import Action

PLACE CALL – SIDE 1

RECEIVE CALL – SIDE 2
Receives a call from Side 1 and sends a message to Side 2.

**MESSAGES: IAM (InitialAddress.HDL)**

**MTP3 Layer:**
- Service Indicator: 0000
- DPC: 00010010 ..010000
- OPC: 01...... 0000010 ....0010

**ISUP Layer:**
- Circuit Identification Code: 00000101 ....00 (5)
- Message Type: 00000001
- Called Party Number: 9880098800
- Calling Party Number: optional parameter

**Initial Address:**
- InitialAddressExport.txt

**Scripts:**
- send "InitialAddress" "InitialAddressImport";
- recv "InitialAddress" "InitialAddressExport";

**Actions:**
1. Recv Action
2. Export Action

**Import Files:**
- InitialAddressExport.txt

**Message Templates:**
- IAM (InitialAddress.HDL)
### ISUP Layer

- **Circuit Identification Code**: 0007
- **Message Type**: 00000001 Initial address
- **Mandatory Variable Parameters**:
  - **Called Address Signal**: 00012
  - **Calling Party Number**: 0012
- **Optional Variable Parameters**:
  - **Calling Address Signal**: 0012
  - **End of optional parameters octet**: x00

### Message Templates

- **Initial Address**: decode "InitialAddressExport"
- **Release**: "Release Complete":
  - **AddressComplete**: decode "InitialAddressExport"
  - **Answer**:

### ISUPMsgHandler

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

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

"Initial Address":
    "Initial Address":
    "AddressComplete":
    "Answer":
    "Release":
    "Release Complete":
    "DefaultMsgHandler":
```

### DATA BUFFER

- **CIC** = 
- **CalledNumber** = 
- **CallingNumber** = 

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

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

"Initial Address":
    "Initial Address":
    "AddressComplete":
    "Answer":
    "Release":
    "Release Complete":
    "DefaultMsgHandler":
```
Send Command With Impairment
CLI/APIs for Remote Control and Test Automation
CLI for Remote Call Control & 3rd Party Integration

load mapsclientc.dll
maps connect 1 192.168.10.32 10024
maps cmd 1 {start "TestBedDefault.xml"}
maps cmd 1 startscript 1 {"Script.gls" "Profile.xml"}
1 # ts =1, cardno = 2

Network
(Analog, & IP Networks)

TCL
Python
Java
VB Scripts

MAPS Client

MAPS Server 1

MAPS Server n

TCP/IP

T1 E1
GL Server

TCP/IP

T1 E1
GL Server

MAPS IP Server 1

MAPS IP Server n
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

```tcl
# source StartTestBed_Subscriber.tcl
Connection Established
Test Bed Started
& Run Place_Call.tcl "Send_Digits" ""
Starting Place call Script...
Script Started
Line 1 Placing Call....
ISDN Call Initiated
Line 1 Waiting For Response....
ISDN Call Proceeding
Line 1 Waiting For Call Connection....
ISDN Call Connected
Sending Btrmf Digits From Line1...
IDM Digits Sent
Line 1 Waiting For Call Disconnect....
ISDN Call Not Disconnected
Line 1 Disconnecting Call....
ISDN Call Completed
Script Stopped
```
Sample SIP Registration VB Script

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>regasm /codebase C:\Users\VBscript\MAPI\Client\CSAPI.dll
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.33440
for Microsoft .NET Framework 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
Microsoft (R) 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>.
```

```
MapsCLI (SIP IETF)

View Latest Command
2015-7-8 10:55:59.430000 : Start "TestBedDefault.xml";
2015-7-8 10:55:13.234000 : LoadProfile "UA_IPv4_Profiles.xml"
2015-7-8 10:55:01.420000 : StartScript 1 "SipRegistrationControl.gci" "Profile0001.1";
2015-7-8 10:55:04.323000 : UserEvent [ "SIP_RegISTRATION" ];
2015-7-8 10:55:54.091000 : StopScript 1;
2015-7-8 10:55:54.392000 : Stop TestBedSetup;
```
Java Client
Python Client
Remote MAPS™ Server

Remote MAPS™
(PKS111 and PKS113)

Client Systems

Multiple Remote MAPS™
Clients connected to single
MAPS™ server system

MAPS Remote Client

MAPS Remote Client

Single Remote MAPS™
Client connected to Multiple
MAPS™ server system

MAPS Remote Client

Server Systems

PC 1
MAPS Core
Listener

PC 2
MAPS Core
Listener

PC 3
MAPS Core
Listener

MAPS Remote Client

MAPS Remote Client

MAPS Remote Client

MAPS Remote Client
Features

- 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
Thank You