iteloffice
computer telephony software

5 Enmore Gardens
London
SW14 8RF

www.iteloffice.com
e: support@iteloffice.com

Tel: +44 (0) 20 8878 7367
Fax: +44 (0) 20 8876 7257

Cuba COM API

Developer Guide

Cuba Version 1.3

Document Number:	**0403-00295**
Revision:	**A**
Date:	**25th May 2009**

0. PRELIMINARIES

0.1 PUBLISHED BY

Itel Office Software Limited, 5 Enmore Gardens, London, SW14 8RF Tel: +44(0)20 8878 7367, Fax: +44(0)20 8876 7257

0.2 COPYRIGHT

0.3 DOCUMENT HISTORY

First issued @ revision A, on 2008-05-25, by A. Comber

BIBLIOGRAPHY

TABLE OF CONTENTS

1. INTRODUCTION

This guide is designed to give you an in depth understanding of installing and integrating the Cuba telephony server developing custom CTI applications for a wide variety of computer telephony requirements.

Cuba is a client server solution. A Cuba Server connects via some form of CTI link to a telephone switch. The Cuba API is a client component which can run local to the Cuba Server or over a LAN or WAN link.

2. OVERVIEW

Providing a Cuba based CTI solution should be approached in the following manner:

1. Install and configure any required CTI link components for specific telephone switch.

2. Decide which is most appropriate Cuba Server type and install and configure for customer CTI link.

3. Provide your custom telephony application on each users workstation.

3. CTI LINK

Currently, Cuba Server supports TAPI and TSAPI CTI link types. Generally, if the telephone switch is an Avaya G3, Definity or S87xx model select The TSAPI Cuba Server type. Otherwise the TAPI Cuba Server.

3.1 TSAPI CTI LINK

The TSAPI Cuba Server communicates with the Avaya telephone system via a TSAPI Server, which must be fully installed and configured.

The TSAPI server is an Avaya product. This software is usually provided by the customer telephone system maintainer.

Avaya historically provided a network interface via a MAPD card. Later models provide an interface via the Avaya Communication Manager hardware device. Whichever methods is used, a TSAPI Server is required by Cuba. The MAPD card or Communication Manager device takes ASAI input / output. TSAPI Server converts the ASAI protocol messages to CSTA messages required by Cuba server.

Avaya 8700, 8500 and 8300 Media Servers provide a network interface via Avaya Communication Manager. Older Avaya systems such as Definity G3 provide a network interface via a MAPD card. TSAPI Server can be provided via Avaya Application Enablement Services, Avaya Computer Telephony 1.2 or 1.3 or Avaya CentreVu CT. Your system maintainer should be able to assist in checking you have the required hardware and software. Email support@iteloffice.com should you require assistance.

Create an account on the TSAPI Server which has full privileges to monitor all the telephony devices which your telephony client application is required to monitor and/or control.

For example, an account called ctiuser with a designated password which has full access privileges to the relevant extensions should be setup. The only other piece of information required is network name of the TSAPI device. This name is NOT the hostname or IP address. A typical TSAPI Server might have the following name: AVAYA#CSTASERV#CSTA#TSAPISERV where TSAPISERV is the hostname of the TSAPI Server.

The Cuba Server installation copies a file, TSLIB.INI to the install folder. The default install folder is **C:\Program Files\Ioffice\Cuba.** Edit this file by entering the IP address of the TSAPI Server. If Cuba server is installed on the same computer as the TSAPI server you can enter 127.0.0.1 as the IP address.

Example where IP address of TSAPI Server is 192.168.0.50:

```
[Telephony Servers]
192.168.0.50=450
```

After editing, copy this file to the computers Windows directory. This is usually C:\Windows. On some computers it may be C:\WINNT or a custom location.

Cuba Server communicates with a TSAPI server on port 450. Ensure that network port 450 is open between Cuba Server and TSAPI Server to permit communication.

3.2 TAPI CTI LINK

Refer to http://www.iteloffice.com/pbx_list.html for specific information on configuring a TAPI CTI link for your telephone system.

3.3 OPERATING SYSTEM REQUIREMENTS

The Cuba Server should run on an ‘always on’ PC on the network.

System Requirements are:

512MB RAM
1.2GHz CPU
Ethernet Port
Operating System – Microsoft Windows: NT, Vista, XP, 2000, 2003, NT

Note that a client application written using the Cuba API can run on any Microsoft Windows platform from Windows 95+.

Administering Cuba Settings

Cuba’s settings are configured in the Cuba Web Administrator - a web interface to Cuba Server. Your web browser must support Java Script.

Your Cuba (Server) communicates Ports 6090 and 8080. Port 6090 is used for communication of telephony data and port 8080 is used for web administration. So you should make sure that these ports are not used by other applications.

Also, if you have a firewall installed between the Cuba (Server) and Cuba (clients) also ensure that it is not blocking ports 6090 and 8080.

3.4 CUBA CLIENT

The Cuba Client, ie your Cuba API application, communicates with the Cuba (Server) on port 6090* so this port must be open. The Cuba (Client) communicates with the Remedy (Client) on each of your workstations and must be installed on each.

* This can be edited in the 'Telephony Server Port' field under 'Server Settings' in the Cuba Web Administrator.

4. INSTALLATION

This section covers the basics of downloading and installing your Cuba Software and introduces you to the Cuba Web Administrator - the web administration panel where you can modify your settings. The Web Administrator is covered in depth in Section 6.

Download and installation of Cuba is fast and intuitive but we've covered it here anyway in a step-by step guide.

4.1 TRIAL PERIOD & LICENCING

During the trial period your Cuba (server) software restricts your operation of Cuba for Remedy to three hours of continuous use. After three hours Cuba Server will shut down but can be restarted for another three hours.

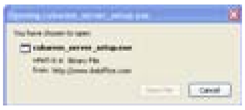
You can apply for a licence by filling in the application form under 'Unlock you Trial' in the Cuba **Web Administrator** – or contact us direct: support@iteloffice.com

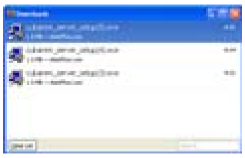
5. DOWNLOADING CUBA

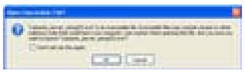
Download your Cuba Software at: **www.iteloffice.com**

To get going with Cuba you need to first download the software. The download operates intuitively and takes just a couple of minutes – but just for the record here's a simple visual overview – using Firefox as the web browser…

Log onto the designated Windows NT/2000/XP/Vista/2003 PC as an administrator.



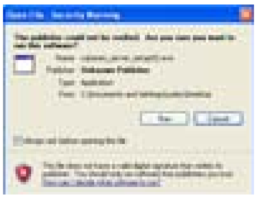


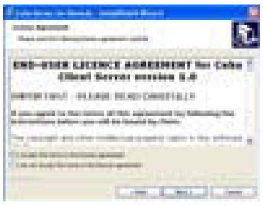


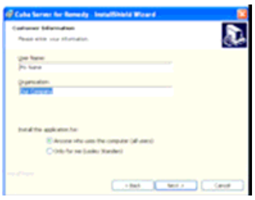
click ‘Save File ‘ >> double click on details >> click ‘OK’ >>

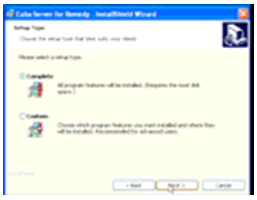
5.1 INSTALLING CUBA (SERVER)

Then the InstallShield Wizard will launch - this will first install the Cuba Server. In this section we will first install the Cuba (Server) then return to install the Cuba (Client).

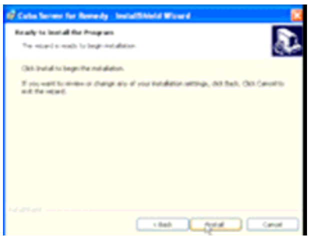


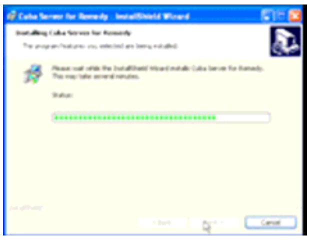


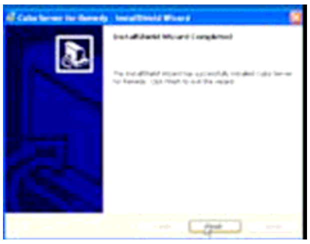




Click Next >> accept licence >> enter name >> click Next >>

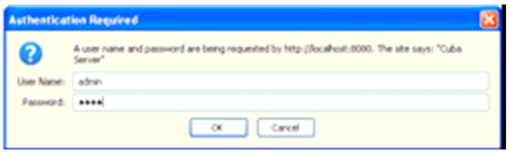






Click Next >> installs >> click finish >>

A dialogue box will appear…



Authentication Required

A user name and password are being requested by http://localhost:8080. The site says: "Cuba Server"

User Name: admin

Password: ****

Enter:

User Name: admin
Password: cuba

Click ‘OK’ and the Cuba Web Administrator launches automatically on the ‘Settings’ page. You can start entering your settings immediately but for the purpose of this tutorial we’ll return to install the Cuba (Client) and deal with settings in the next Section.

5.2 INSTALLING THE CUBA (CLIENT)

The only required client component is the Cuba COM dll, CubaCOM.dll, which must be registered on each client workstation. CubaCom.dll can be registered by issuing the command: regsvr32 cubaCOM.dll

6. THE CUBA WEB ADMINISTRATOR

You can control Cuba 's Server settings in the Cuba Web Administrator.

This is a web front end that allows quick and easy configuration of your Cuba (Server).

The Web Administrator launches automatically during initial installation and opens on the 'Settings' tab. This gives you a series of simple form fields where you can enter data.

6.1 NAVIGATING TO THE WEB ADMINISTRATOR

You'll be able to return here to review and edit all of your settings whenever you need to. You'll find the Cuba Web Administrator alongside the Cuba Software Files in your Program File - simply click on:

Start > All Programs > Cuba > Cuba Web Administrator > or

If your Cuba (Server) is installed on your local machine you can use the address 127.0.0.1.

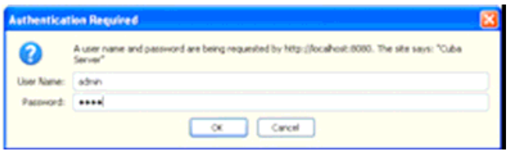
If your Cuba (Server) is installed elsewhere on your network, launch your web browser and navigate to the hostname or IP address of the computer where your Cuba (Server) is installed.

Append : 8080 to the hostname, for example: **http://MyComputerName:8080**

This is because the Cuba web server uses the non standard port 8080 so as not to conflict with alternative web servers which may be configured on the Cuba (Server) computer.

6.2 PASSWORD PROTECTION

Each time you open it a dialogue box will appear…



A user name and password are being requested by http://localhost:8080. The site says: "Cuba Server"

User Name: admin

Password: ••••

As before enter:

User Name: admin

Password: cuba

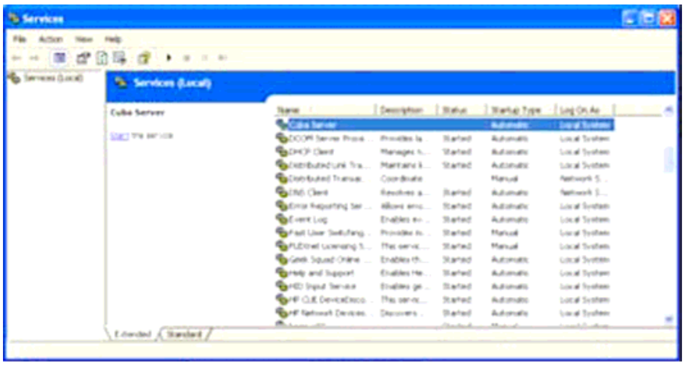
Click ‘OK’ and the Cuba Web Administrator launches automatically on the ‘Settings’ page.

If you get a Page Load Error then either the Cuba (Server) is not hosted on your machine or your trial version needs to be restarted.

Go to:

Start > Control Panel > Administrative Tools > Services >

Scroll Down to ‘Cuba Server’ > Click ‘Start’



Cuba Server

Start the service

Name	Description	Status	Startup Type	Log On As
Cuba Server			Automatic	Local System
DCOM Server Proce...	Provides la...	Started	Automatic	Local System
DHCP Client	Manages n...	Started	Automatic	Local System
Distributed Link Tra...	Maintains li...	Started	Automatic	Local System
Distributed Transac...	Coordinate...		Manual	Network S...
DNS Client	Resolves a...	Started	Automatic	Network S...
Error Reporting Ser...	Allows erro...	Started	Automatic	Local System
Event Log	Enables ev...	Started	Automatic	Local System
Fast User Switching ...	Provides m...	Started	Manual	Local System
FLEXnet Licensing S...	This servic...	Started	Manual	Local System
Geek Squad Online ...	Enables th...	Started	Automatic	Local System
Help and Support	Enables He...	Started	Automatic	Local System
HID Input Service	Enables ge...	Started	Automatic	Local System
HP CUE DeviceDisco...	This servic...	Started	Automatic	Local System
HP Network Devices...	Discovers ...	Started	Automatic	Local System

The Cuba Server will restart. You may need to restart your browser in order to access the Cuba Web Administrator.

7. SETTINGS

7.1 CUBA LOGS

Select the Logs section to view or delete Cuba Server logs. The log file provides a history of what is happening on the Cuba Server.

7.2 DEVICES

In this section you need to enter all the call centre agents extension numbers. Not the agent ID, the actual telephone hardware/handset extension number. The Cuba client extension logon dialog is populated with the contents of the list defined here.

7.3 QUEUES

In this section you need to enter the queues into which call centre agents login to. Cuba monitors the queues specified in this section at startup. So if this setting is changed it is necessary to restart Cuba Server for the changes to take effect. As agents login and logout of these queues, Cuba intercepts the logon and logoff events and retains an internal mapping between devices (agent phone) and AgentID.

7.4 CTI LINK

Enter the TSAPI Server name, TSAPI User name and TSAPI password as discussed in section **Error! Reference source not found.**.

7.5 HELP

Help on configuring Cuba Server.

7.6 UNLOCK TRIAL

To get a full user licence you will need to purchase an Activation Key. Enter your contact details here and click 'Send Registration Request'. We will contact you to discuss payment options and the registration process.

7.7 REGISTER

Once you have received your Activation Key copy and paste it here and click 'Save'. You will then get unrestricted use of Cuba for the number of seats that you have purchased.

8. TROUBLESHOOTING

8.1 LOGGING

Cuba Server writes logging information to a log file, cubasvr.txt, in the Itel subdirectory of the common application data folder. The location of this folder varies dependent on Windows version. On Windows XP this folder is likely to be C:\Documents and Settings\All Users\Application Data\Itel.

For full logging you need to run Cuba Server as a command line process rather than as a service. Follow these steps:

1. Stop the Cuba Server service. Start… Control Panel, Administrative Tools, Services. Double click on Cuba Server entry and click Stop button.

2. Open up a command prompt. Start… All Programs… Accessories, Command Prompt.

3. Navigate to C:\Program Files\IOffice\Cuba\ and enter following command followed by enter: CubaSvc.exe -debug

4. Do not close the console window.

Now debug level errors and informational messages will be logged to cubasvr.txt which can provide very useful troubleshooting information.

9. THE API

9.1 METHODS

```
        [id(1), helpstring("method ServerConnect")] HRESULT ServerConnect([in]
BSTR host, [in] BSTR port, [out, retval] long* retCode);

        [id(2), helpstring("method ServerDisconnect")] HRESULT
ServerDisconnect([out, retval] long* retCode);

        [id(3), helpstring("method Register")] HRESULT Register([in] BSTR device,
[out, retval] long* retCode);

        [id(4), helpstring("method Unregister")] HRESULT Unregister([in] BSTR
device, [out, retval] long* retCode);

        [id(5), helpstring("method MakeCall")] HRESULT MakeCall([in] BSTR device,
[in] BSTR destination, [in] BSTR userdata, [out, retval] long* retCode);

        [id(6), helpstring("method ClearCall")] HRESULT ClearCall([in] BSTR device,
[in] long connectionid, [out, retval] long* retCode);

        [id(7), helpstring("method AnswerCall")] HRESULT AnswerCall([in] BSTR
device, [in] long connectionid, [out, retval] long* retCode);

        [id(8), helpstring("method HoldCall")] HRESULT HoldCall([in] BSTR device,
[in] long connectionid, [out, retval] long* retCode);

        [id(9), helpstring("method RetrieveCall")] HRESULT RetrieveCall([in] BSTR
device, [in] long connectionid, [out, retval] long* retCode);

        [id(10), helpstring("method InitiateTransfer")] HRESULT
InitiateTransfer([in] BSTR device, [in] BSTR destination, [in] long connectionid, [in]
BSTR userdata, [out, retval] long* retCode);

        [id(11), helpstring("method CompleteTransfer")] HRESULT
CompleteTransfer([in] BSTR device, [in] long heldconnid, [in] long consultconnid,
[out, retval] long* retCode);

        [id(12), helpstring("method InitiateConference")] HRESULT
InitiateConference([in] BSTR device, [in] BSTR destination, [in] long connectionid, [in]
BSTR userdata, [out, retval] long* retCode);

        [id(13), helpstring("method CompleteConference")] HRESULT
CompleteConference([in] BSTR device, [in] long heldconnid, [in] long consultconnid,
[out, retval] long* retCode);
```

[id(14), helpstring("method SetForward")] HRESULT SetForward([in] BSTR device, [in] BSTR destination, [in] long fwdmode, [out, retval] long* retCode);

[id(15), helpstring("method CancelForward")] HRESULT CancelForward([in] BSTR device, [in] long fwdmode, [out, retval] long* retCode);

[id(16), helpstring("method AttachUserData")] HRESULT AttachUserData([in] BSTR device, [in] long connectionid, [in] BSTR userdata, [out, retval] long* retCode);

[id(17), helpstring("method AgentLogin")] HRESULT AgentLogin([in] BSTR device, [in] BSTR queue, [in] BSTR agentid, [in] BSTR agentpassword, [in] long agentworkmode, [out, retval] long* retCode);

[id(18), helpstring("method AgentLogout")] HRESULT AgentLogout([in] BSTR device, [in] BSTR queue, [out, retval] long* retCode);

[id(19), helpstring("method AgentSetReady")] HRESULT AgentSetReady([in] BSTR device, [in] BSTR queue, [in] long workmode, [out, retval] long* retCode);

[id(20), helpstring("method AgentSetNotReady")] HRESULT AgentSetNotReady([in] BSTR device, [in] BSTR queue, [in] long workmode, [out, retval] long* retCode);

[id(21), helpstring("method GetDeviceList")] HRESULT GetDeviceList([in, out] BSTR* devices, [in] BSTR userdata, [out, retval] long* retCode);

[id(22), helpstring("method GetNumDevices")] HRESULT GetNumDevices([in, out] long* numdevices, [out, retval] long* retCode);

[id(23), helpstring("method GetPrivateData")] HRESULT GetPrivateData([in] BSTR privateid, [in, out] BSTR* data, [out, retval] long* retCode);

[id(24), helpstring("method SetPrivateData")] HRESULT SetPrivateData([in] BSTR privateid, [in] BSTR data, [out, retval] long* retCode);

9.2 EVENTS

[id(1), helpstring("method Alerting")] void Alerting([in] long timestamp, [in] long errorcode, [in] int calltype, [in] int role, [in] long callid, [in] long connid, [in] int addressid, [in] long requestid, [in] BSTR thisDN, [in] BSTR otherDN, [in] BSTR CallerID, [in] BSTR CalledID, [in] BSTR digits, [in] BSTR userdata, [in] BSTR agentid);

[id(2), helpstring("method Connected")] void Connected([in] long timestamp, [in] long errorcode, [in] int calltype, [in] int role, [in] long callid, [in] long connid, [in] int addressid, [in] long requestid, [in] BSTR thisDN, [in] BSTR otherDN, [in] BSTR CallerID, [in] BSTR CalledID, [in] BSTR digits, [in] BSTR userdata, [in] BSTR agentid);

[id(3), helpstring("method Cleared")] void Cleared([in] long timestamp, [in] long errorcode, [in] int calltype, [in] int role, [in] long callid, [in] long connid, [in] int

```
addressid, [in] long requestid, [in] BSTR thisDN, [in] BSTR otherDN, [in] BSTR
CallerID, [in] BSTR CalledID, [in] BSTR digits, [in] BSTR userdata, [in] BSTR agentid);

            [id(4), helpstring("method Initiated")] void Initiated([in] long timestamp,
[in] long errorcode, [in] int calltype, [in] int role, [in] long callid, [in] long connid, [in]
int addressid, [in] long requestid, [in] BSTR thisDN, [in] BSTR otherDN, [in] BSTR
CallerID, [in] BSTR CalledID, [in] BSTR digits, [in] BSTR userdata, [in] BSTR agentid);

            [id(5), helpstring("method Error")] void Error([in] long timestamp, [in] long
errorcode, [in] int calltype, [in] int role, [in] long callid, [in] long connid, [in] int
addressid, [in] long requestid, [in] BSTR thisDN, [in] BSTR otherDN, [in] BSTR
CallerID, [in] BSTR CalledID, [in] BSTR digits, [in] BSTR userdata, [in] BSTR agentid);

            [id(6), helpstring("method Registered")] void Registered([in] long
timestamp, [in] long errorcode, [in] long requestid, [in] BSTR thisDN);

            [id(7), helpstring("method Unregistered")] void Unregistered([in] long
timestamp, [in] long errorcode, [in] long requestid, [in] BSTR thisDN);

            [id(8), helpstring("method CTILinkConnected")] void CTILinkConnected([in]
long timestamp, [in] long errorcode);

            [id(9), helpstring("method CTILinkDisconnected")] void
CTILinkDisconnected([in] long timestamp, [in] long errorcode);

            [id(10), helpstring("method ServerConnected")] void ServerConnected([in]
long timestamp, [in] long errorcode);

            [id(11), helpstring("method ServerDisconnected")] void
ServerDisconnected([in] long timestamp, [in] long errorcode);
```