



Avaya Solution & Interoperability Test Lab

Application Notes for configuring Avaya Proactive Contact R5.1 with CallScripter R4.5 – Issue 1.0

Abstract

These Application Notes describe the configuration steps required for CallScripter R4.5 to successfully interoperate with Avaya Proactive Contact R5.1.

Information in these Application Notes has been obtained through DevConnect compliance testing and additional technical discussions. Testing was conducted via the DevConnect Program at the Avaya Solution and Interoperability Test Lab.

1. Introduction

These Application Notes describe a compliance tested configuration comprised of Avaya Proactive Contact R5.1 using Computer Telephony Integration (CTI) (also known as soft dialer) and CallScripter R4.5.

Avaya Proactive Contact R5.1 uses Avaya Aura® Application Enablement Services to connect Avaya Proactive Contact Agent headsets on Avaya Aura® Communication Manager with outbound calls.

Avaya Proactive Contact Agents log into Proactive Contact using the intuitive CallScripter web interface. The dialer selection, login credentials, job selection, tenant selection, agent state, call control options and all other agent interactions are presented to the user by the CallScripter.CTI.ProactiveContact.dll toolbar provider which communicates with the Avaya Proactive Contact R5.1 Agent API via moagent32.dll.

The CallScripter Application can be used in either hard-dialer or soft-dialer CTI environments and can identify the appropriate functionality to offer the agent depending on the dialer environment capabilities. CallScripter can handle inbound, outbound, blend and managed jobs, is able to support multi-dialer, multi-tenant and multi-unit work list configurations.

CallScripter is configured so that the appropriate pre-configured Script is presented to the agent according to the job through which outbound calls are delivered to the agent. The script can be fully customized according to customer needs and can present a variety of options and pages through which the agent navigates during each outbound call transaction.

2. General Test Approach and Test Results

The interoperability compliance testing evaluated the ability of CallScripter to carry out call handling functions in a variety of scenarios through the Application Programming Interface (API) with Proactive Contact. The feature test cases were performed automatically. Outbound calls were automatically placed and delivered to CallScripter by Proactive Contact. Different types of jobs were exercised, along with a variety of actions initiated from CallScripter, to verify proper generation and handling of supported messages from Proactive Contact. The Proactive Contact Editor was used to start/stop and configure jobs. The verification included checking the display of fields, options, and values on CallScripter, and verification of the exchanged API events in the agentx_API.trans log files was also performed. All test cases were executed.

DevConnect Compliance Testing is conducted jointly by Avaya and DevConnect members. The jointly-defined test plan focuses on exercising APIs and/or standards-based interfaces pertinent to the interoperability of the tested products and their functionalities. DevConnect Compliance Testing is not intended to substitute full product performance or feature testing performed by DevConnect members, nor is it to be construed as an endorsement by Avaya of the suitability or completeness of a DevConnect member's solution.

2.1. Interoperability Compliance Testing

The feature testing focused on verifying successful login of Proactive Contact Agents using CallScripter and the display of CallScripter Scripts with appropriate options, fields, and values for the following scenarios:

- Outbound and managed jobs
- Select dialer, select tenant, login, join job, select unit work list, go on/off break, change job, leave job, and logoff
- Place managed call, cancel managed call, agent drop, release line, and finish work
- Set recall and Agent Owned Recall and update customer fields
- Set completion codes

2.2. Test Results

All test cases were completed successfully with the following observations:

- Where a shadow job is used for recalls the correct script failed to pop. This was due to the dynamic nature under which shadow jobs are created and the field used in the AORNotify message to trigger the Script pop. e.g. Proactive Contact will send:

AGTAORNotify NAgent server **shadowjob_2Allidoutbnd2** to the agent.

Where shadowjob_2 is the job dynamically created for the recalls and outbnd2 is the original job which would be linked to a Script to pop.

CallScripter.CTI.ProactiveContact.dll v1.0.0.4 resolves this issue.

2.3. Support

Support for the CallScripter product can be obtained as follows:

- Website: <http://www.callscripiter.com>
- Email: helpdesk@callscripiter.com
- Phone: +44 (0)844 544 8882

Reference Configuration

Figure 1 below shows the compliance tested configuration comprising of Communication Manager hosted on an S8800 Media Server connected to a G450 Media Gateway. PRI trunks from the G450 Media Gateway provide the path for outbound calls to the simulated PSTN. Proactive Contact Agents are logged into Proactive Contact using the web based CallScripter interface and 96xx IP Deskphones provide the agent headsets. A TSAPI connection between Proactive Contact and Application Enablement Services is used to connect place outbound calls and connect to agent headsets.

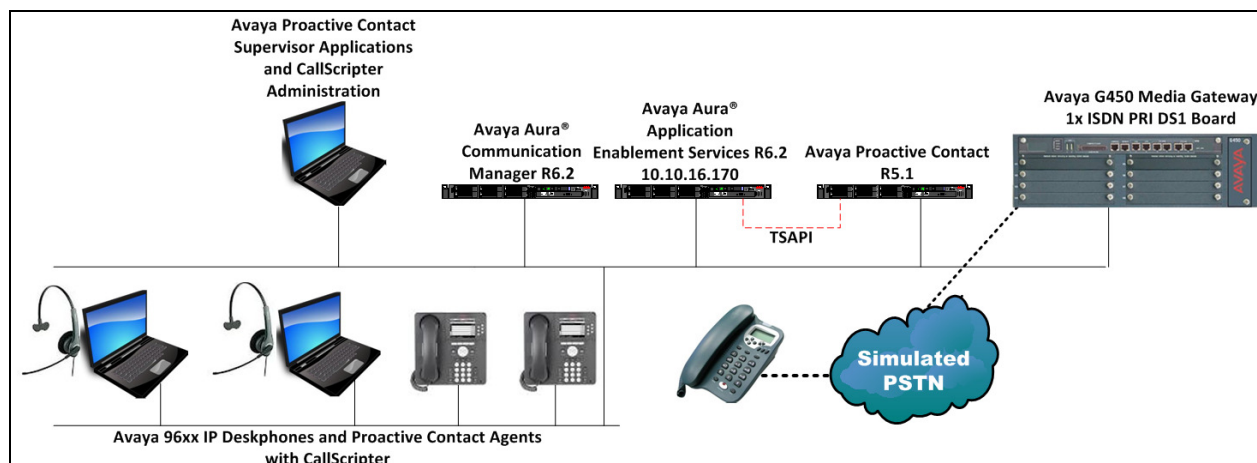


Figure 1: Avaya Proactive Contact with CallScripter Configuration

3. Equipment and Software Validated

The following equipment and software were used for the sample configuration provided:

Equipment/Software	Release/Version
Avaya Aura® Communication Manager running on Avaya S8800 Server	R6.2 SP6
Avaya Proactive Contact Running on HP DL360	R5.1.0.0.1802
Avaya Aura® Application Enablement Services VE vAppliance	R6.2 patch 1
Avaya 9630 IP Deskphone	3.2
Avaya PG230	Generic Software 15.3.1
CallScripter running on Windows Vista PC	CallScripter Environment 4.5.24.5654 CallScripter.CTI.ProactiveContact.dll 1.0.0.0

4. Configure Avaya Aura® Communication Manager

The Application Notes assume that the procedures for configuring Communication Manager to support the Proactive Contact CTI integration have been completed. This includes the correct licensing and configuration of agent headsets, AEServices, CTI Links and adjunct route VDNs and announcements. See **Section 10** for more information.

5. Configure Avaya Aura® Application Enablement Services

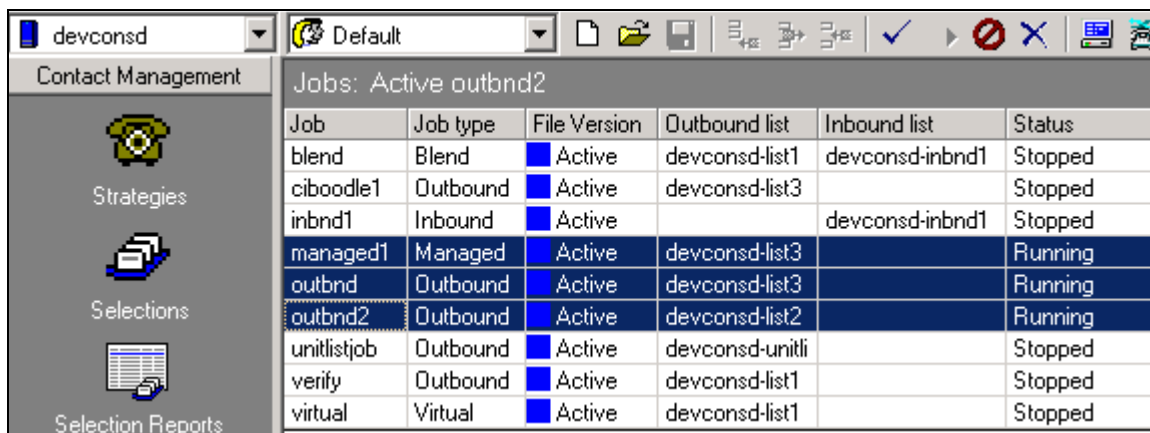
The Application Notes assume that the procedures for configuring Application Enablement Services to support the Proactive Contact CTI integration have been completed. This includes the correct licensing and configuration of Switch Connections, TSAPI Links and CT Users. See **Section 10** for more information.

6. Configure Avaya Proactive Contact

It is assumed a fully operating Proactive Contact has been implemented. If assistance is required with Proactive Contact, please contact Avaya Professional Services or an Avaya Business Partner. The following pages illustrate the configuration of the appropriate Proactive Contact jobs used for the compliance test.

6.1. Perform Avaya Proactive Contact Job Configuration

It is assumed that the necessary pre-configuration of relevant job components such as tenants, calling lists, strategies, record selections and jobs have already been configured as required.. For the purpose of the compliance testing various job types were configured named **outbnd**, **outbnd2**, and **managed1** as shown below in the Proactive Contact Editor application.



The screenshot shows the Proactive Contact Editor application interface. On the left is a sidebar with icons for 'Contact Management', 'Strategies', 'Selections', and 'Selection Reports'. The main area displays a table titled 'Jobs: Active outbnd2'. The table has columns for Job, Job type, File Version, Outbound list, Inbound list, and Status. The 'outbnd2' job is highlighted in blue.

Job	Job type	File Version	Outbound list	Inbound list	Status
blend	Blend	Active	devconsd-list1	devconsd-inbnd1	Stopped
ciboodle1	Outbound	Active	devconsd-list3		Stopped
inbnd1	Inbound	Active		devconsd-inbnd1	Stopped
managed1	Managed	Active	devconsd-list3		Running
outbnd	Outbound	Active	devconsd-list3		Running
outbnd2	Outbound	Active	devconsd-list2		Running
unitlistjob	Outbound	Active	devconsd-unitli		Stopped
verify	Outbound	Active	devconsd-list1		Stopped
virtual	Virtual	Active	devconsd-list1		Stopped

7. Configure CallScripter

The CallScripter application is deployed and configured according to customer requirements, through engagement directly with the CallScripter Project Team. The configuration steps required for integration to Proactive Contact can be summarized as follows:

- Configure windows service connection to Avaya Proactive Contact
- Link Job to Script
- Link Job Data to CallScripter Script Fields
- Create CallScripter User
- Configure agent toolbar connection to Avaya Proactive Contact

7.1. Configure Connection to Avaya Proactive Contact

The CallScripter application contains a Windows Service that polls the Avaya environment(s) to retrieve a list of tenants, active jobs, data fields and outcomes (completion codes). This information is stored locally within the CallScripter SQL database and used to connect the CallScripter scripts to the relevant Jobs/Fields. All communication between CallScripter and the Avaya Proactive Contact dialler is through the Avaya Agent API (moagent32.dll).

The windows service is part of the CallScripter installation process and is configurable via configuration keys within the CallScripter system.

Navigate to the CallScripter installation directory and using an appropriate text editor open **CSEFProactiveContactDataLoader.dll.config** and edit the configuration keys as follows:

- **MosaixPort** – enter **22700** which is the default port used by the Proactive Contact Agent API
- **MosaixUserName** – enter an appropriate Proactive Contact username in this case **agent5**
- **MosaixPassword** – enter a corresponding password
- **MosaixRetryTime** – enter an appropriate polling interval, in this case **1800000**

```
<!--
  Setting:      MosaixPort
  Valid values: positive whole number
  Description:  Port number to connect to proactive contact
-->
<add key="MosaixPort" value="22700"/>

<!--
  Setting:      MosaixUserName
  Valid values: text
  Description:  valid user to connect to proactive contact with
permissions to read and connect to all jobs
-->
<add key="MosaixUserName" value="agent5"/>

<!--
  Setting:      MosaixPassword
  Valid values: text
  Description:  password for above username
-->
<add key="MosaixPassword" value="05"/>

<!--
  Setting:      MosaixLogPath
  Valid values: text
  Description:  log for framework to write text logs
-->
<add key="MosaixLogPath" value="C:\Program
Files\CallScripter\Utilities\EnterpriseFramework\logs"/>

<!--
  Setting:      MosaixRetryTime
```

```
Valid values: positive whole number
Description: Thread sleep time in ms between sourcing data.
Typically 30 minutes : 1800000
-->
<add key="MosaixRetryTime" value="1800000"/>
```

7.2. Link Job to Script

After logging into the CallScripter application, users with the relevant licences applied to their profiles will be able to gain access to the CallScripter script editor section of the main menu.



Within the Script Editor admin screens, the user will choose a customer and a script to administer (or create a new customer/script). In order for a Script to pop when a call is delivered to an agent for a specific job, the Script and job must be linked within the CallScripter application. Using the CallScripter Script Editor, select the appropriate job to link to in the **Campaign** drop-down box (these drop downs will be populated by the windows service). The user will need to drill down through dialer, tenant, job type and select the job they wish to link to the script from the **Campaigns** drop down box. By clicking on the **Add Link** button and clicking **Save** (not shown) the link will be established.

The screenshot displays the CallScripter Script Editor interface. On the left, a sidebar titled 'CallScripter' contains a search bar and a list of scripts: 'Avaya Inbound', 'Avaya Outbound' (highlighted), 'Outbound Test Script', and 'Outbound Script'. Below the list are buttons for 'Add', 'Copy', 'Export', 'Import Version', and 'Delete'. The main area is titled 'Script Editor' and includes sections for 'Automatic version Cleanup' (with an 'Enabled' checkbox), 'Script Type' (with radio buttons for 'Inbound Calling Script', 'Outbound Calling Script' (selected), 'Non-Calling Script', 'Email Script', and 'Component Script Beta'), '+ Script Permissions', '+ Administer', and '- Campaign Links'. The 'Campaign Links' section contains dropdowns for 'Server' (10.10.10.90), 'Tenant' (Default), 'Campaign Type' (Managed), and 'Campaigns' (Please choose...). The 'Campaigns' dropdown is highlighted with a red box, and its options are 'Please choose...', 'Please choose...', and 'managed1'. A red box also highlights the 'managed1' option. Below the dropdowns, a note states: '* Denotes that the campaign is currently assigned to another script. Assigning this campaign to this script will remove it from any other script.' An 'Add Link' button is present. At the bottom, a 'Selected Campaigns' section shows 'managed1' and 'NiceBlend' with 'ACCTNUM' dropdowns and red 'X' icons. The bottom status bar indicates 'Logged in as: Andy (Andy)'.

Once the link has been created between the job name and the script the field which should be used as the reference for the script pop must be configured This is usually set to **ACCTNUM** but any field in the job can be used as a reference field providing it is unique for each record in the calling list.

- Campaign Links:

Server

10.10.16.90

Tenant

Default

Campaign Type

Outbound

Campaigns

outbnd2

* Denotes that the campaign is currently assigned to a script. Assigning this campaign to this script will remove it from any other script.

Add Link

Selected Campaigns:

outbnd2

ACCTNUM

outbnd3

ACCTNUM

outbnd4

ACCTNUM

unitlistjob

ACCTNUM

Save Changes

View Script History

View Script Runs

Note: The reference field chosen must be unique for each record in the call list.

7.3. Link Job Data to CallScripter Script Fields

Once a Proactive Contact job has been linked to a CallScripter Script, a list of Calling List data fields associated to the Proactive Contact job will be presented in the Script Editor, as an attribute for each field in the script. The Proactive Contact Calling List data fields are linked to fields within the Script so that on delivery of a call to the agent the appropriate script will pop based on the link to the job, and the calling list detail will be displayed to the agent by the Script in the relevant fields on the page. This link will act as a two-way communication between the script field and the Proactive Contact Calling List data fields. If the agent updates information in the script and transitions off the Script page (or closes the Script) the data for each field that is linked will be automatically written back to the Proactive Contact Calling List data fields. Multiple Script fields can be linked to the same Proactive Contact Calling List data field. If this is the case, the last field to be rendered on the page will be the one that applies its data to the call list on page transition.

The screenshot below displays the Script **Avaya Outbound (2)** in the Script Editor with the Calling List data fields presented on the left hand side in the control attributes.

The screenshot displays the CallScripter Script Editor interface. On the left, a sidebar titled 'Field Link' lists various Calling List data fields. The main editor area shows the script fields for 'Avaya Outbound (2)', including text boxes for account number, title, forename, and surname, as well as dropdowns, text areas, and checkboxes. A red 'Next' button is located at the bottom right of the editor area.

Select the appropriate **Field Name** in this case **Account Number** and from the **Field Link** drop down box select the field in the Calling List to link to, in this case **ACCTNUM**.

The screenshot displays a configuration window titled "Account Number". It contains several sections: "Field Information" with a "Field Name" text box containing "Account Number" and a "Default Value" text box; "Field Help" with an empty text box; and "Field Link" with a dropdown menu. The dropdown menu is open, showing a list of fields. "ACCTNUM" is highlighted in blue. Other visible fields in the list include FIFTSTAT2, BALANCE, TOTALDUE, NAME1, NAME2, CITY, STATE, ZIPCODE, PHONE1, PHONE2, COMMENT1, AGENT, DTE, TME, CODE, TME_STAMP, DIALERID, ABDNOTE, ABDNTME, ABDNCODE, JOBNAME, COUNTER, ENTRYDATE, STATUSFLAG, RECALLNAME, RECALLDATE, RECALLTIME, ORIGINALJOBNAME, and NAME.

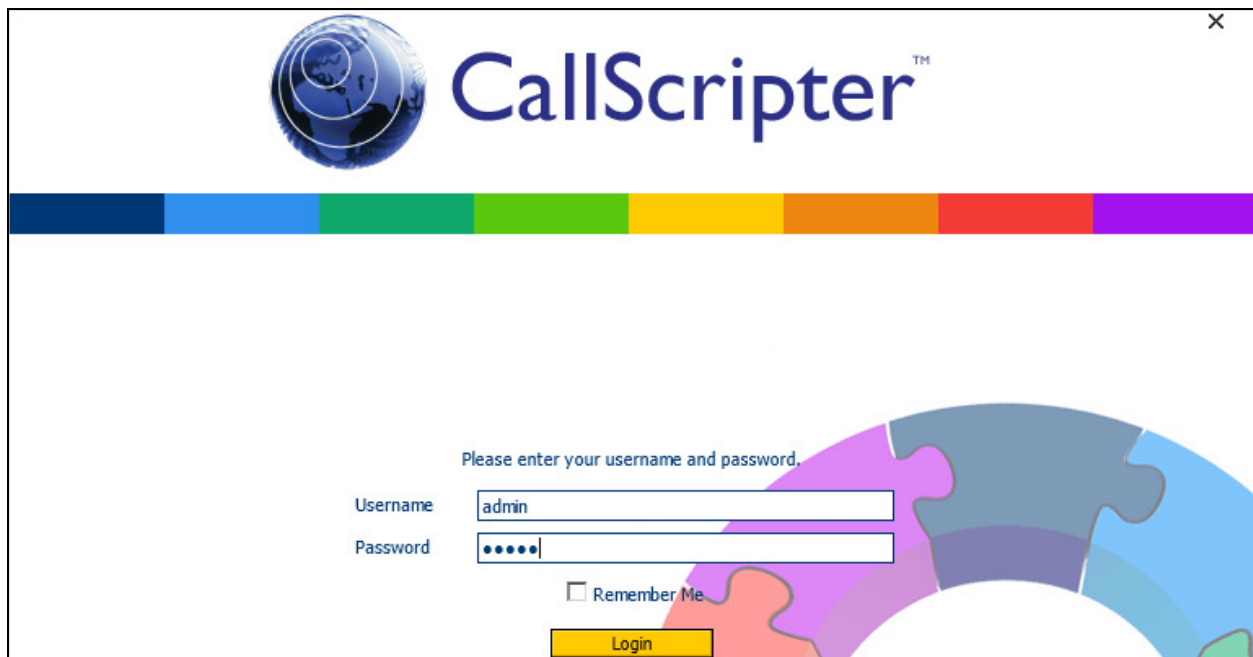
Field Name	Field Link
Account Number	ACCTNUM

7.4. Create CallScripter User

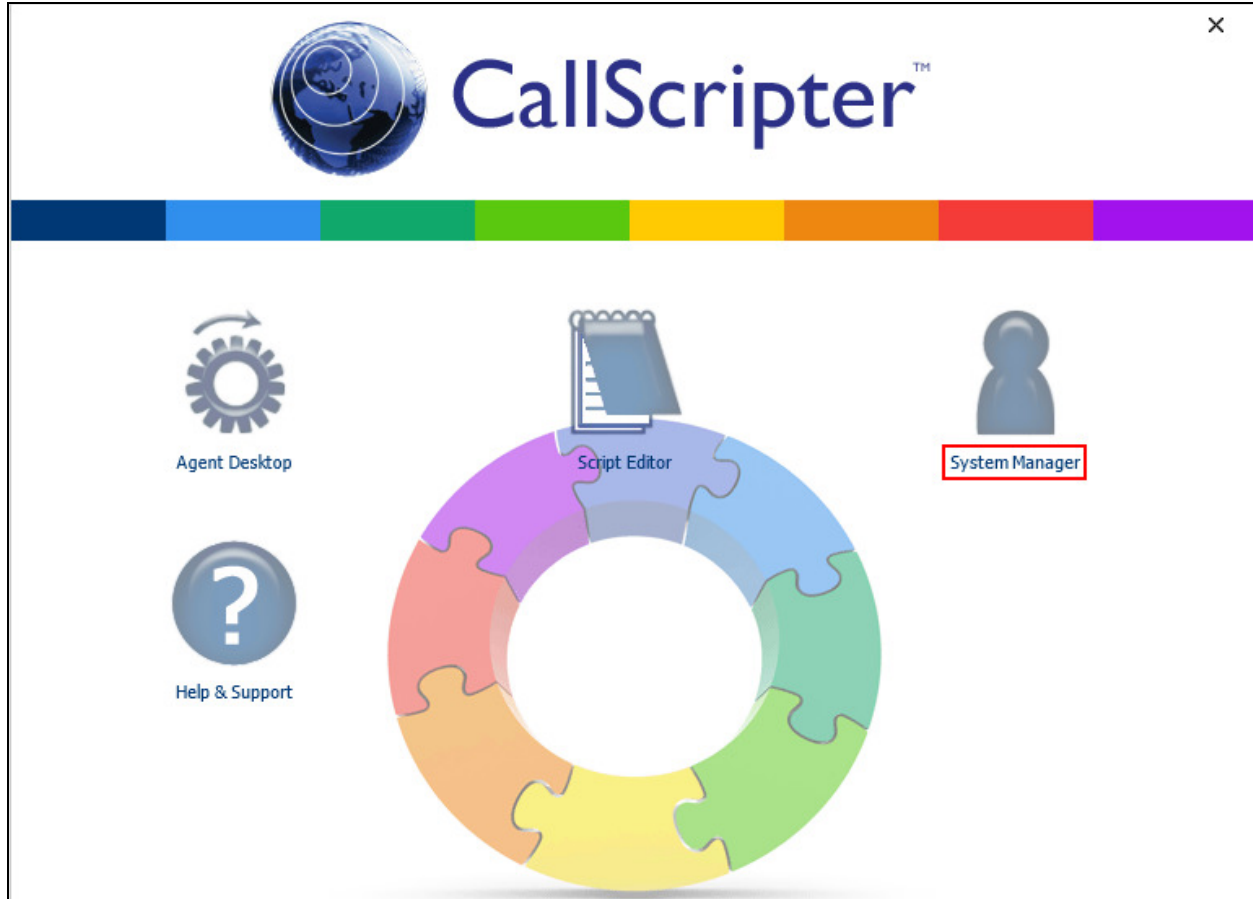
A user must be configured within the CallScripter application to allow an agent to log into CallScripter and start taking/making calls using the Avaya Proactive Contact integration. The username created in CallScripter doesn't have to match the usernames in Proactive contact.

The user will access the CallScripter application via Internet explorer by navigating to the CallScripter Main page in this case <http://callscripwin7/?ra=true> and login using the user credentials (username and password). In order to add users to the application the user must login using a profile that has the admin licence applied to it.

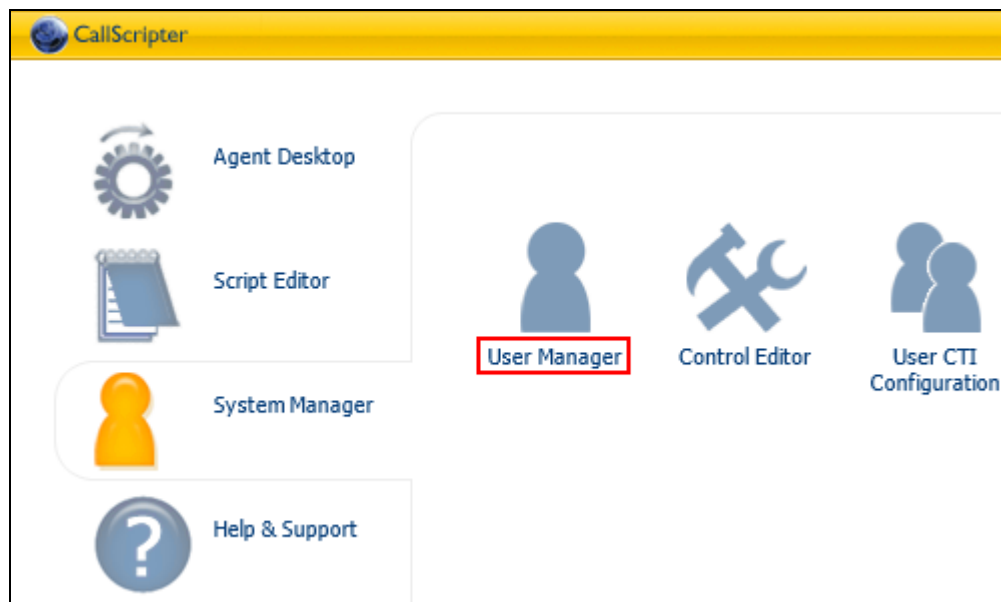
Note: If CallScripter is setup to run in windows authentication mode, the username within CallScripter must match the windows username that the agents are logging onto the PC with.

The image shows the CallScripter login page. At the top, there is a logo consisting of a blue globe with white lines, followed by the text "CallScripter™". Below the logo is a horizontal bar with seven colored segments: dark blue, light blue, green, yellow, orange, red, and purple. The main content area has a light gray background. In the center, there is a login form. Above the form, the text "Please enter your username and password." is displayed. The form has two input fields: "Username" with the text "admin" and "Password" with five dots. Below the password field is a checkbox labeled "Remember Me". At the bottom of the form is a yellow button labeled "Login". To the right of the login fields, there is a large, colorful graphic of interlocking puzzle pieces in various colors (blue, green, yellow, orange, red, purple, pink, and light blue).

The following screen will appear, click **System Manager** as shown below.



Click **User Manager**.



In the **Add new:** field, enter a new username and click **Add**.

The screenshot displays the CallScripter User Manager interface. On the left, a sidebar contains tabs for 'Users', 'Groups', and 'Licensing'. Below these tabs, a list of users is shown: 'Admin User (Admin)', 'Andy (Andy)', and 'Richard (Richard)'. At the bottom of this list, the 'Add new:' field is highlighted with a red box, containing the text 'Ricardo'. To the right of this field is an 'Add' button. The main area of the interface is titled 'User Manager' and contains a form for adding a new user. The form includes fields for 'Username' (admin), 'Firstname' (Admin), 'Surname' (User), 'Enabled' (checked), 'Language' (English (United States)), 'Time Zone' (System Default Time), and 'Password'. A 'Group Membership' section on the right shows a checked box for 'System'.

Enter a **Password** and click **Save**.

CallScripter

Users Groups Licensing

Please select a user:

+ [dropdown]

Admin User (Admin)
Andy (Andy)
Ricardo (Ricardo)
Richard (Richard)

Add new:
[text box]

Add Import

☐ Show disabled users

User Manager

User Details:

Username: Ricardo
Firstname: Ricardo
Surname: [text box]
Enabled: ☒
Language: Default (English) [dropdown]
Time Zone: System Default Time [dropdown]
Password: [masked]

Group Membership:

☒ System

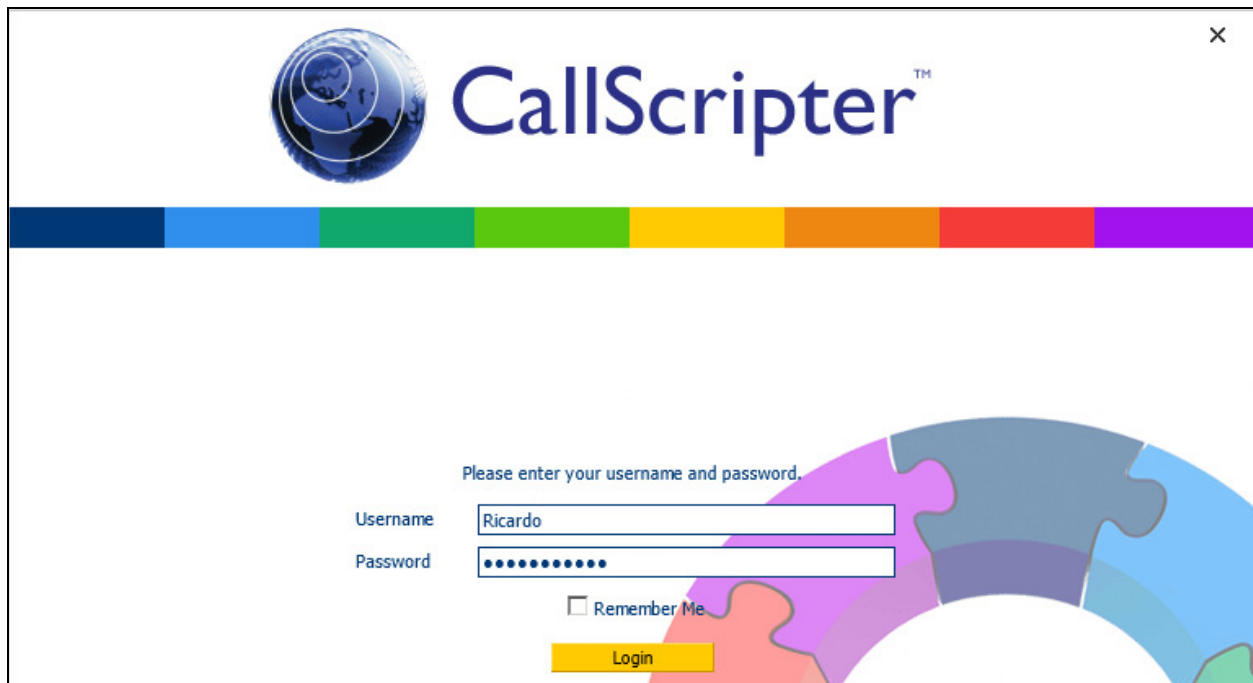
Save

Once a user has been added to CallScripter they will automatically be given the licence to access the **Agent Desktop** within CallScripter. Should additional licences be needed to allow the user to access other parts of the application, they can be applied using the licensing tab in the top left corner of the screen.

Note: If the user needs the ability to link Scripts to Proactive Contact jobs or amend Scripts they will need to be assigned the **Script Editor** licence to see this part of the application.

7.5. Login New CallScripter User

From the main login page login using the newly created CallScripter User.



The image shows the CallScripter login page. At the top, there is a logo consisting of a blue globe with white lines and the text "CallScripter™" in blue. Below the logo is a horizontal bar with eight colored segments: dark blue, light blue, green, lime green, yellow, orange, red, and purple. The main content area is white. In the center, there is a login form. Above the form, the text "Please enter your username and password." is displayed. The form has two input fields: "Username" with the text "Ricardo" and "Password" with masked characters (dots). Below the password field is a checkbox labeled "Remember Me". At the bottom of the form is a yellow "Login" button. The background of the page features a large, colorful puzzle piece graphic on the right side.

Please enter your username and password.

Username

Password

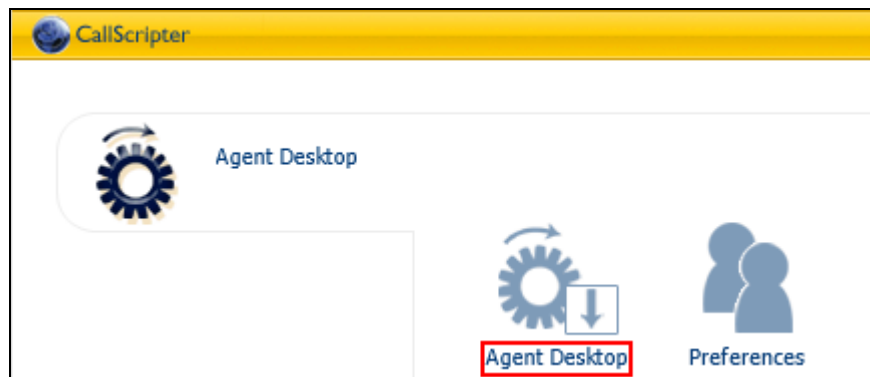
☐ Remember Me

Login

Click **Agent Desktop**.



And click **Agent Desktop** again.



The following page will appear, in the Toolbar select the appropriate **Host** address, enter the correct **Tenant**, Proactive Contact agent **User Name** and corresponding **Password** and the **Headset ID** of the agent (in this case the agents desk phone) and click **Login**.

The screenshot shows the 'Login to Proactive Contact' dialog box. The fields are filled as follows:

- Host:** 10.10.16.90
- Tenant:** Default
- User Name:** agent6
- Password:** (masked with dots)
- Headset ID:** 6017

A yellow 'Login' button is located at the bottom right of the dialog. The background application window shows the 'CallScripter' logo and a toolbar with options like 'Log', 'Offline', and 'Login to Proactive Contact'. A search bar is visible at the bottom right of the application window.

Proactive Contact will place a call using CTI to extension 6017 and play the pre-recorded Welcome to Proactive Contact message.

Once logged in successfully the user will be prompted to select a job type and job name from the Toolbar.

The screenshot shows the 'Jobs' menu open in the CallScripter application. The menu options are:

- Outbound
- Inbound
- Managed
- Blended
- ReList

The 'Outbound' option is selected, and a sub-menu is visible with the following options:

- outbnd
- outbnd2

The background application window shows the 'CallScripter' logo and a toolbar with options like 'Log', 'No Work', 'Actions', and 'Jobs'. A search bar is visible at the bottom right of the application window.

If the job is configured to require unit ID for agent login, then the user will also be prompted to select a unit name on the toolbar, else the user will be attached to the job without this step.

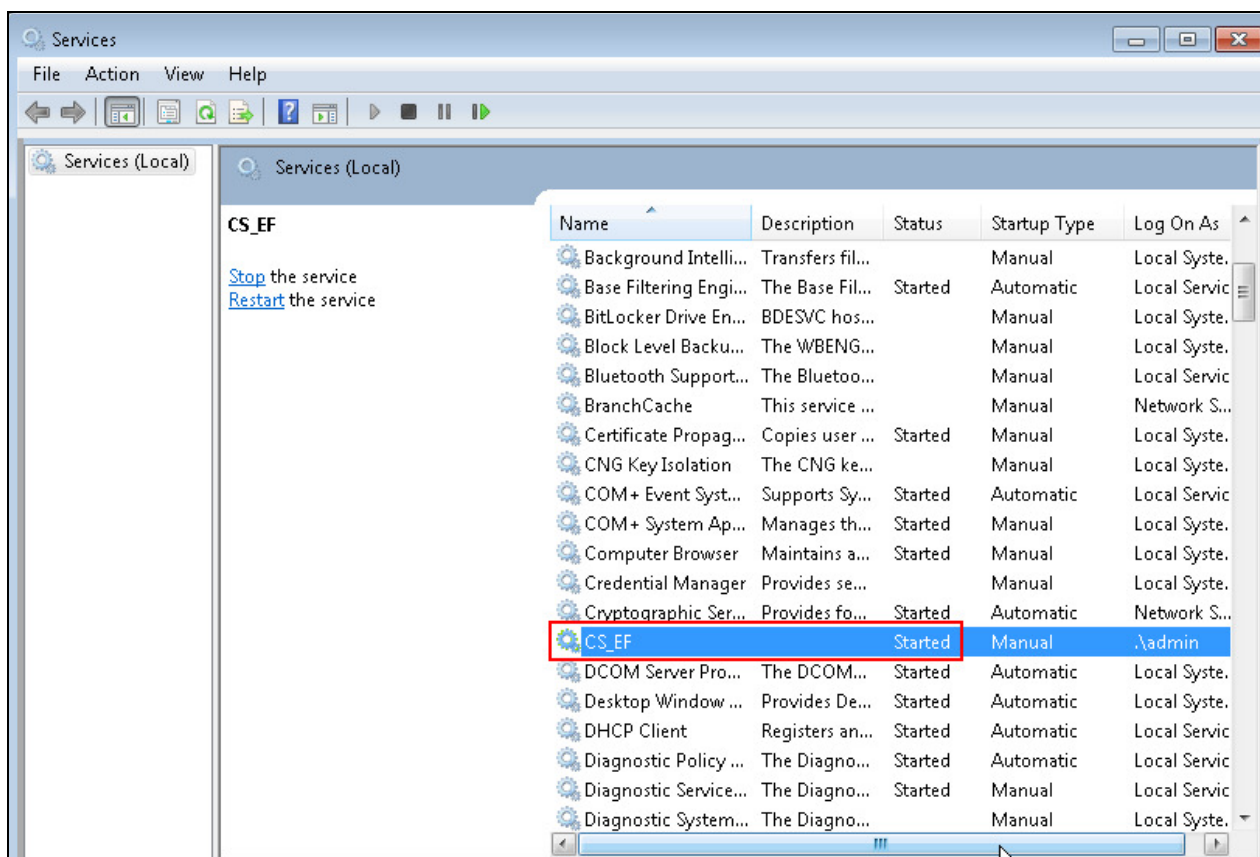
At this stage the agent can use the intuitive Toolbar to perform agent duties and call control as usual.

8. Verification Steps

This section provides the tests that can be performed to verify the proper configuration of CallScripter with Proactive Contact. Prior to verification, start an appropriate job on Proactive Contact.

8.1. Verify Jobs and Call List Data are polled by CallScripter

Verify the CallScripter Proactive Contact windows service has started; load the Windows Services snap-in and ensure **CS_EF** has a status of **Started**.



The Service will begin to poll the Avaya servers for active jobs, fields and outcomes. To verify that this service is working correctly the windows service log file can be analyzed for activity. Navigate to the pre-configured logs directory and open the latest .log file, in this case **C:\Program Files\CallScripter\Utilities\EnterpriseFramework\logs\2013\07\03.log**. Verify that the log file displays events pertaining to the successful login to the relevant Proactive Contact **Server**, the **Tenant** selection, the **Active Job** listing and **Datafields** and **outcome codes** for that job.

```
05:00:02.878 - Sourcing campaign data
05:00:05.163 - Initialising Object
05:00:05.403 - Current Server: 10.10.16.90
05:00:05.405 - Current Tenant: Default
05:00:05.407 - LOGGING ON
05:00:06.975 - Active job: outbnd2
05:00:07.280 - 107 Datafields found for: outbnd2
05:00:07.866 - 13 outcome codes found for: outbnd2
05:00:18.165 - Detached from job
05:00:18.168 - Active job: outbnd
05:00:18.474 - 107 Datafields found for: outbnd
05:00:19.165 - 13 outcome codes found for: outbnd
05:00:29.465 - Detached from job
05:00:29.466 - Active job: managed1
05:00:29.769 - 107 Datafields found for: managed1
05:00:30.065 - 13 outcome codes found for: managed1
05:00:40.401 - Detached from job
05:00:40.406 - LOGGING OFF
05:00:40.414 - Logged off Event:
05:00:40.417 - releasing MosServer....
05:00:40.423 - releasing DllConfig....
05:00:40.425 - releasing SerStart....
05:00:40.430 - Initialising Object
05:00:40.722 - LOGGING ON NEW TENANT - Ten01
05:00:42.065 - LOGGING OFF
05:00:42.070 - Logged off Event:
05:00:42.071 - releasing MosServer....
05:00:42.108 - releasing DllConfig....
05:00:42.144 - releasing SerStart....
```

Once the service has successfully sourced the jobs, the relevant fields will begin to populate these options within CallScripter Editor as demonstrated in **Section 7.2**.

- Campaign Links:

Server

10.10.16.90

Tenant

Default

Campaign Type

Outbound

Campaigns

outbnd2

* Denotes that the campaign is currently assigned to a script. Assigning this campaign to this script will remove it from any other script.

Add Link

Selected Campaigns:

outbnd2

ACCTNUM

outbnd3

ACCTNUM

outbnd4

ACCTNUM

unitlistjob

ACCTNUM

Save Changes

View Script History

View Script Runs

8.2. Verify Successful CallScripter Agent Operation

Login a Proactive Contact Agent using CallScripter. Verify a list of jobs is presented, join a job and begin servicing calls. Confirm that the correct Script is presented to the agent and is pre-filled with the correct Calling List data. Using the Toolbar verify that the full variety of call control and agent state options are available and operate as expected.

The screenshot shows the CallScripter interface. At the top, there's a navigation bar with 'Log', 'On Call', and a toolbar with icons for home, back, forward, and search. A 'Call Control' dropdown menu is open, showing options: 'Hangup Call', 'Hold/Resume', 'Dialing options', 'Transfer options', 'Adjust Headset', and 'Do Not Call'. The main content area is titled 'Demonstration 1: Passing data between the script and the dialer'. It contains a text box with the following text: 'In this example the various types of script fields below are linked to the columns in the call list. When the script is presented to the agent the data associated with populate the data in the fields that have been linked to the dialer call list. If the agents change this information within the script, the data will be automatically save data when the script page transitions/refreshes.'

Below the text, there are several form fields:

- Account Number: 5300292221274730
- Title: (empty)
- Forename: JOHN DOE
- Surname: JOHN DOE
- Address1: (empty)
- Address2: (empty)
- Town / City: (empty)
- County / State: (empty)
- Postcode / ZIP: 7221
- Find Address: (button)

Below the form fields, there are several other controls:

- Dropdown: (empty)
- Textbox: (empty)
- Textarea: (empty)
- Check boxes: ☐ Checkbox 1 ☐ Checkbox 2 ☐ Checkbox 3
- Radio Buttons: ☐ Radio Button 1 ☐ Radio Button 2 ☐ Radio Button 3

8.3. Verify Avaya Aura® Communication Manager CTI Link Status

Enter the command **status aesvcs cti-link** and verify that the **Service State** is **established** for the relevant **CTI Link**.

status aesvcs cti-link						
AE SERVICES CTI LINK STATUS						
CTI Link	Version	Mnt	AE Services	Service State	Msgs Sent	Msgs Rcvd
1	4	no	aes62vm	established	20	20

8.4. Verify Avaya Proactive Contact Job Status

From Proactive Contact shell, type the command **jobmon** to verify agent is logged into the job outbnd2 and handling a call.

```
[STANDARD]                               Job Activity
[ALLID]

                               Summary Statistics
Job: [outbnd2][1769]
Start time: 12.38.03   Current time: 15.45.26

Agent Activity                               Line Usage
-----
All Outb   ACD      PTP   Outbound Lines      Cur  Avg  Peak
Logged in: 1      1      0      0   Demand      :      1    0    1
Assigned  : 1      1      :      Available    :      9
On Phone  : 1      1      :      Total Lines   :      10

Calling Activities
-----
Outbound Phone Calls
Records Selected:      386
Phone Calls made:      4
Cur/Run Hit Rate:      65/1  %
Agent Connects   :      0
Queue            :      0
Recalls          :      2
Phone Calls Left:      378

[ Job outbnd4 ready for calling ]
```

8.5. Verify Avaya Aura® Application Enablement Services CT User Status

From the Application Enablement Services OAM web interface, click **Status → Status and Control → TSAPI Service Summary**, and click on the **User Status** button (not shown) and verify that the CTI user configured for Proactive Contact is listed.

Status | Status and Control | TSAPI Service Summary Home

CTI User Status

☐ Enable page refresh every 60 seconds

CTI Users: All Users Submit

Open Streams 4
Closed Streams 13

Open Streams

Name	Time Opened	Time Closed	Tlink Name
pc51hd	Fri Jul 12 13:06:46 2013		AVAYA#CM62#CSTA-S#AES62VM
pc51sd	Mon Jul 22 09:54:47 2013		AVAYA#CM62#CSTA-S#AES62VM
pc51sd	Mon Jul 22 09:55:13 2013		AVAYA#CM62#CSTA-S#AES62VM
DMCCLCUserDoNotModify	Mon Jun 24 14:21:31 2013		AVAYA#CM62#CSTA#AES62VM

Show Closed Streams Back

9. Conclusion

These Application Notes describe the configuration steps required for CallScripter to successfully interoperate with Avaya Proactive Contact in a CTI configuration. All feature test cases were completed successfully with observations noted in **Section 2.2**.

10. Additional References

This section references the product documentation relevant to these Application Notes.

1. *Administering Avaya Proactive Contact*, Release 5.1, April 2013, available at <http://support.avaya.com>.
2. *Avaya Aura® Application Enablement Services Administration and Maintenance Guide*, Release 6.2, Issue 1, July 2012, available at <http://support.avaya.com>.
3. *Sample Avaya Proactive Contact 3.0 with CTI Installation and Configuration – Issue 1.0*, 6/21/2006

©2013 Avaya Inc. All Rights Reserved.

Avaya and the Avaya Logo are trademarks of Avaya Inc. All trademarks identified by ® and ™ are registered trademarks or trademarks, respectively, of Avaya Inc. All other trademarks are the property of their respective owners. The information provided in these Application Notes is subject to change without notice. The configurations, technical data, and recommendations provided in these Application Notes are believed to be accurate and dependable, but are presented without express or implied warranty. Users are responsible for their application of any products specified in these Application Notes.

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya DevConnect Program at devconnect@avaya.com.