Mechanism for performing temporary handoff to IVR and return to agent

Introduction

TBC

Mechanism

TBC

Implementation

The following section describes the configuration steps required to add the temporary handoff mechanism to an existing UCCE/CVP deployment already configured for CVP Comprehensive call flows with CVP defined as a Type 10 VRU.

1 Ingress gateway

- Copy cvp_ivrhandoff.tcl to flash on the ingress gateway
- Add the cvp_ivrhandoff service as follows:

```
application
    service cvp_ivrhandoff flash:cvp_ivrhandoff.tcl
```

- Add optional parameters to the service as required

ivrhandoff-dnis-prefix	Digit prefix for IVR handoff DNIS. (Default: 487). The VoiceXML gateway must be configured to route calls with this prefix to this ingress gateway.
ivrhandoff-agent-hold-media	Media to play to agent while caller and IVR legs are connected. (Default: flash:holdmusic.wav).
ivrhandoff-validate-guid	Enable validation of the IVR call leg header IVR-Handoff-Guid against the agent leg guid to prevent IVR spoofing. (Default: true).

- To get started quickly I would suggest setting the parameter *ivrhandoff-validate-guid* to false as it keeps the ICM scripting as simple as possible.
- Ensure the service is loaded using the command

```
call application voice load cvp_ivrhandoff
```


Paul Tindall, Cisco, Version 0.1, 28 February 2012

- Add the cvp_ivrhandoff service to the incoming dial-peer instead of using CVP survivability, for example:

```
dial-peer voice 999001 pots
    service cvp_ivrhandoff
    incoming called-number 651963499
```

2 VoiceXML gateway

- Copy bootstrap_ivrhandoff.tcl to flash on the VoiceXML gateway. This can be the same device as the ingress gateway.
- Add/modify the bootstrap service as follows. Clearly you can opt to keep the original bootstrap service if preferred and use a different name for this one although this modified bootstrap.tcl script does preserve all the standard functionality.

```
application
    service bootstrap flash:bootstrap_ivrhandoff.tcl
```

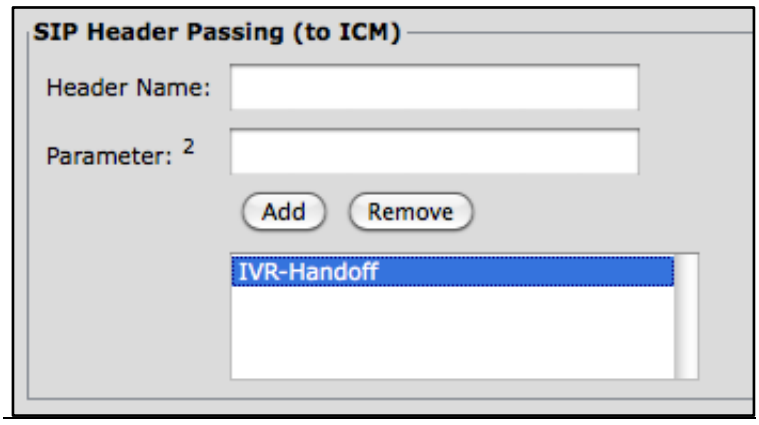
- Ensure the service is (re)loaded using the command

```
call application voice load bootstrap
```

- Configure an outgoing VoIP dial-peer to route the shunt call to the ingress gateway(s). Note that when using multiple ingress gateways, each one should have a its own IVR handoff prefix to ensure that the shunt call is always routed to the ingress gateway the caller is terminated on.

3 CVP call server

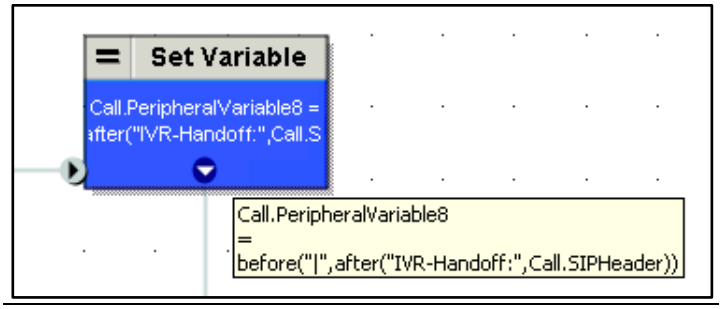
To ensure that the IVR-Handoff SIP header is passed to ICM and available to the script in the SIPHeader variable you must add it to the SIP Header Passing list in CVP via the OAMP console, as shown below.



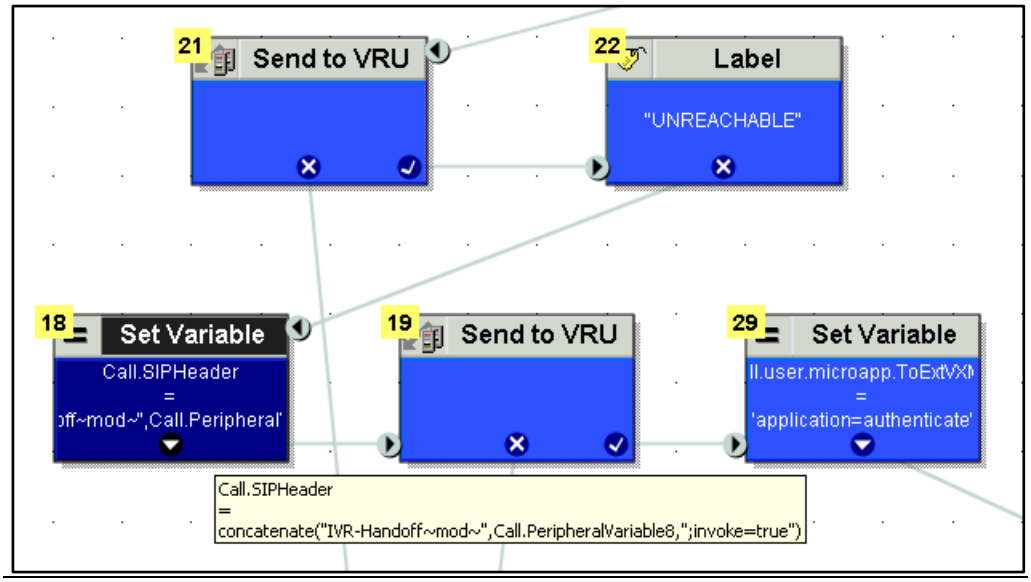
4 ICM scripting

Configuring the main script that handles the incoming call.

Add a script node as shown to extract the IVR-Handoff SIP header into a variable of your choice. It's useful when setting up and testing to use a variable that will be displayed at the agent CTI desktop. In that way, you can easily confirm the SIP header passing is working correctly and the prefix is configured as you intended.



Configuring the post-route script used to set up the consultative call to IVR.

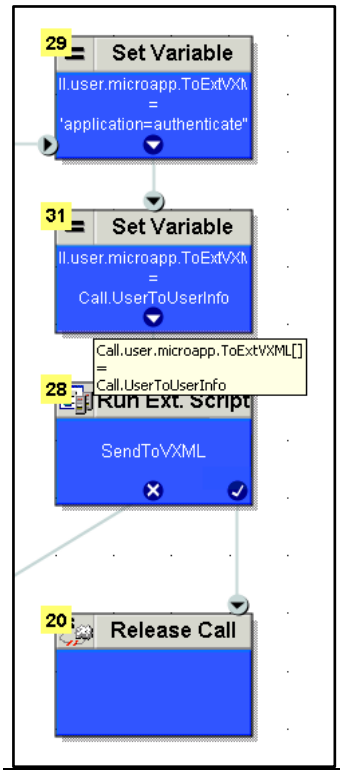


Script Node	Description
21	Set up a call from the agent IP phone to CVP which is configured as a Type 10 VRU. Ensure you have a label for the UCM as a routing client configured on the ICM VRU. Also ensure the UCM dial plan will route the label returned by ICM, remembering that it will include the trailing variable correlation ID. This is the standard configuration required for agent to (Type 10) IVR calls that pass context; it has nothing additional required specifically to support the mechanism described in this document.
22	Tear the current call to VRU down and force a return to the script by delivering a label that is deliberately unreachable. Forcing disconnection of the VRU leg is a necessary workaround in order that we can regain control in the script and set the IVR-Handoff SIP header on the VRU leg. (Note. This is exactly the same workaround that can be used to forcibly remove ASR/TTS resources from a call by destroying the current VRU leg.)
18	Populate the SIPHeader variable with the new IVR-Handoff header that will trigger the handoff mechanism. The required header format is "IVR-Handoff: *<handoff-*

	number>;invoke=true" where <*handoff-number*> is the value extracted and saved by the main incoming script.
19	Now set up a new VRU leg which this time will trigger the shunt call between the VoiceXML and ingress gateways because IVR-Handoff with invoke=true is present.
29	Call has arrived at the VRU, the shunt call has been established, so now invoke the actual IVR application using the standard GS microapp method for CVP VoiceXML applications.

5 Playing progress audio to the agent

So the agent can remain aware of how the caller is progressing through the IVR application and receive an audible outcome at the end, it is possible to play progress messages to the agent under control of the CVP VoiceXML application despite the voice path being connected between the application and the caller. The following instructions describe how to configure this.

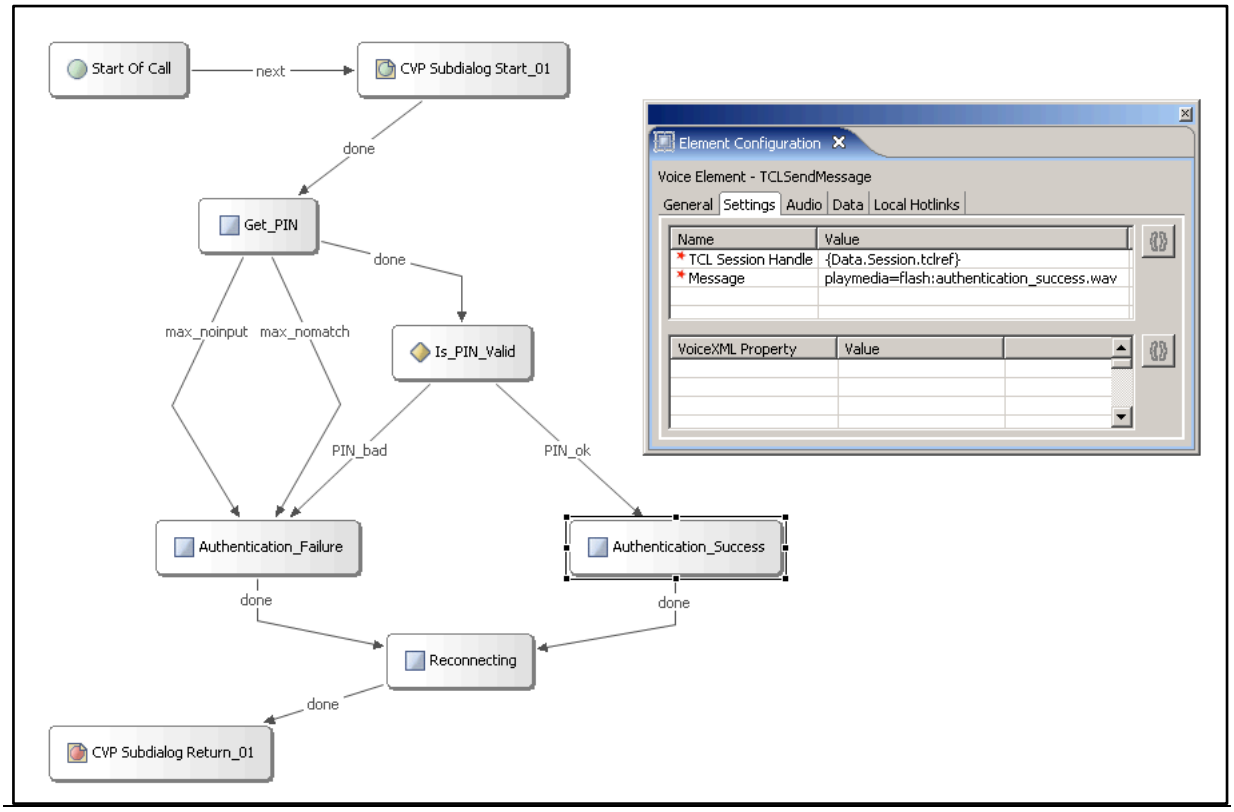


Script Node 31

Pass the contents of the UserToUserInfo variable to the CVP VoiceXML application by populating one of the ToExtVXML array variables.

UserToUserInfo contains the handle of the TCL session to which the agent remains connected during the handoff. Using a custom Call Studio element the CVP VoiceXML application can pass requests to this session to play progress messages to the agent. The TCL session handle is available to the CVP VoiceXML application via the session data item *tclref*. Note that control returns immediately to the CVP VoiceXML application after the request has been sent to the agent's TCL session; the message playing will complete asynchronously and the interaction with the caller will not incur any delays.

The following shows an example CVP VoiceXML application specifically outlining use of the TCLSendMessage custom element to play audio to the agent, in this case following successful authentication. There are two things to note here. Firstly, the media file is specified using an absolute URI in this version and secondly, because control is returned to the application immediately after the authentication-success message is triggered it is also necessary to play something to the caller in parallel. Without this, the application would end immediately before the agent could hear the outcome message. In this example "*Thank you. Please hold while I reconnect your advisor*" is played to the caller.



To make the TCLSendMessage custom element available, simply drop the JAR file into the following locations for Call Studio and CVP VoiceXML Server respectively and restart them.

- C:\Cisco\CallStudio\eclipse\plugins\com.audiumcorp.studio.library.common_8.5.1\lib
- C:\Cisco\CVP\VXMLServer\common\lib

Configure an additional service on the VoiceXML gateway as follows:

- Copy cvp_tclsndmsg.tcl to flash.
- Add the cvp_tclsndmsg service.

```
application
    service cvp_tclsndmsg flash:cvp_tclsndmsg.tcl
```

- Ensure the service is loaded using the command

```
call application voice load cvp_tclsndmsg
```

6 Additional security

To enable GUID validation when the shunt call is established to the dynamic IVR handoff number, simply save the contents of the ICM variable CallGUID in the main incoming call script and add it as the value of the optional *guid* attribute when the IVR-Handoff SIP header is built in the IVR call script, for example:

```
IVR-Handoff: 487336983;invoke=true;guid=4D4186FC5BF411E1AF5BB6C99DD76296
```

Finally, either remove the cvp_ivrhandoff service parameter *ivrhandoff-validate-guid* or set it to true.

Additional considerations

The cvp_ivrhandoff.tcl script has to displace survivability.tcl until such time as the two are merged. It's important therefore to be aware that survivability script features/functionality will be lost at the present time.