



Avaya Solution & Interoperability Test Lab

Application Notes for Servion Call Back Manager with Avaya Interactive Response 1.2 - Issue 1.0

Abstract

These Application Notes describe the interoperability compliance testing of Servion Call Back Manager version 1.1 with Avaya Interactive Response version 1.2. Servion Call Back Manager is a windows-based software product that enables call centers with Avaya Interactive Response the ability to offer queued callers the option to continue to wait in the queue, or to request a call back when an agent of a particular skill set is available.

The interoperability test included installation and testing of this product in a simulated call center environment. Testing concluded that Servion Call Back Manager version 1.1 successfully interoperates with Avaya Interactive Response. Information in these Application Notes has been obtained through compliance testing and additional technical discussions. Testing was conducted via the Developer*Connection* Program at the Avaya Solution and Interoperability Test Lab.

1. Introduction

The function of a call center is to efficiently distribute customer calls to a set of agents. When the volume of customer callers exceeds the number of available agents, calls are maintained in queues to be distributed when agents become available. Avaya Communication Manager has the capability to provide callers in these queues with information such as their expected wait time until a specific skilled agent would be available for their call. Servion Call Back Manager, a Windows-based software application, extends these capabilities by using Avaya Interactive Response to offer queued callers the option to continue to wait in the queue, or to request a call back when an agent of a particular skill set is available.

For Interoperability Compliance Testing, the configuration shown in **Figure 1** was used.

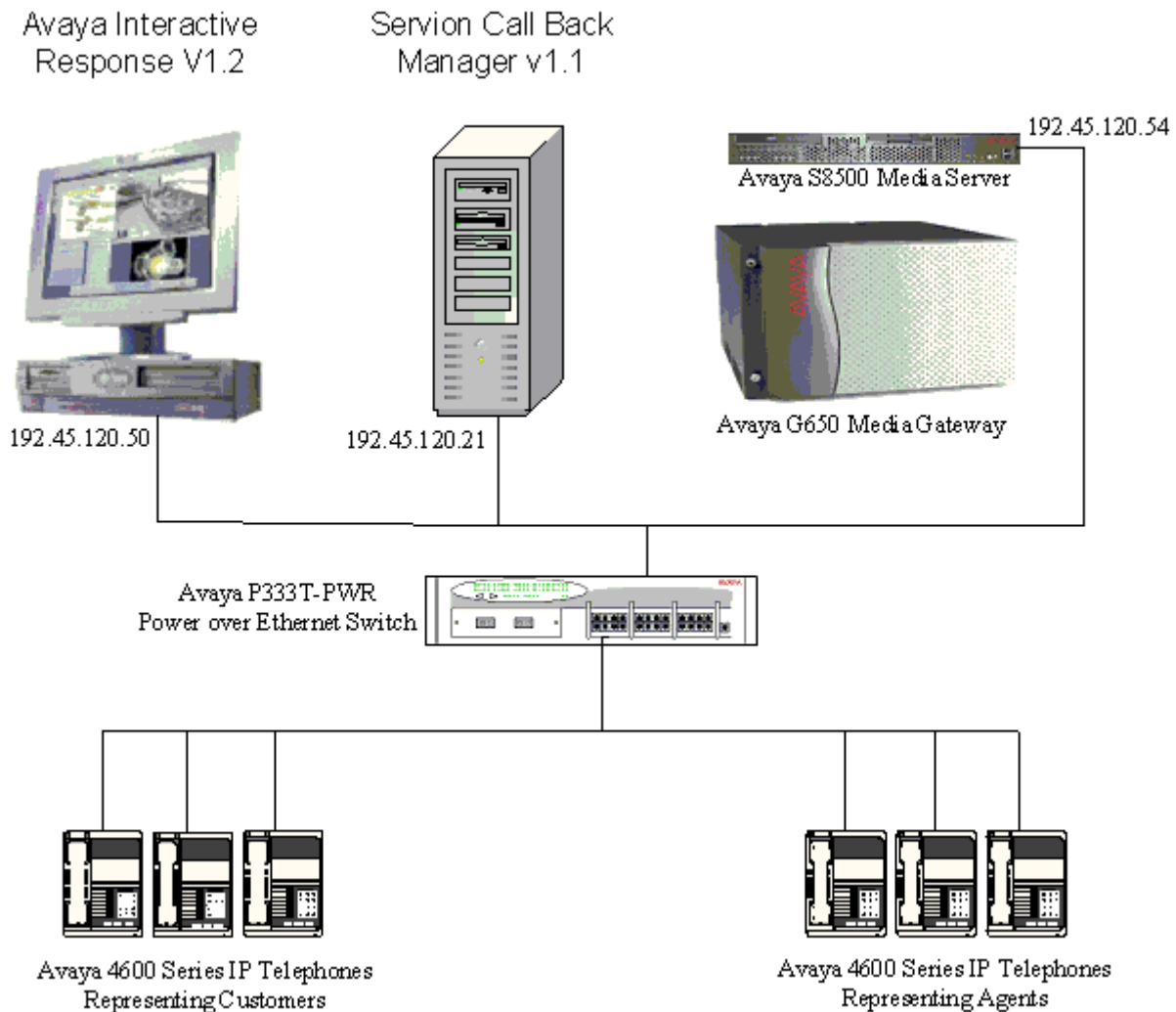


Figure 1: Sample Configuration For Servion Call Back Manager and Avaya Interactive Response

This test configuration has an Avaya Interactive Response connected to an Avaya S8500 Media Server with an Avaya G650 Media Gateway in a typical call center environment. Servion Call Back Manager is installed on a PC with an Oracle Database. This PC can have either Windows 2000 Server or Windows 2003 Server as its operating system. Testing was performed on both of these operating systems.

2. Equipment and Software Validated

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

The Interoperability testing was conducted using two different operating systems, Microsoft Windows 2000 Server and Microsoft Windows 2003 Server. Both were qualified.

Equipment	Software
Avaya Interactive Response 1.2 on SunBlade 120 <ul style="list-style-type: none"> Vonetix 3.6.5 	V1.2.055
Avaya S8500 Media Server with G650 Media Gateway	Communications Manager 2.1 R012x.01.0.411.7
Generic PC <ul style="list-style-type: none"> Pentium IV 1 GB RAM 40 GB+ hard disk 100 Mhz Network interface 	<ul style="list-style-type: none"> Servion Call Back Manager Version 1.1 Oracle 9i Windows 2000 Server w/ MSMQ or Windows 2003 Server w/ MSMQ IIS 5.0 or Higher Internet Explorer 6.0 Microsoft .NET Framework 1.1 with SERVICE PACK 1.0

3. Configuration using Avaya Communication Manager

Servion Call Back Manager uses a Call Vector in Avaya Communication Manager to queue incoming calls to an agent. While callers are waiting in this queue, the **converse-on** vectoring command is used to connect the waiting caller with the Call Back Manager's IVR script running on Avaya Interactive Response. The call vector used in Interoperability Compliance Testing is shown in the following screen capture. Note that with this vector the waiting caller remains queued to the agent's skill at all times. This means that if an agent becomes available while the caller is in the process of scheduling a call back request on Avaya Interactive Response, the caller will immediately be disconnected from Avaya Interactive Response and connected to the newly available agent. In many applications this is the desirable result. In cases where it is not, a different call vector can be used. See reference [2] in Section 8 Additional References. Note that steps 7 through 14 in the following vector were included for Interoperability Testing purposes (so that automated call test equipment could verify that Avaya Communication Manager received the correct call status by looking at the length of the call) and are not needed in field applications.

```

display vector 2                                     Page 1 of 3
CALL VECTOR_
Number: 2                                           Name: Call Back mgr
Basic? y      EAS? y      G3U4 Enhanced? y      Meet-me Conf? n      Lock? n
Prompting? y  LAI? y      G3U4 Adv Route? y      ANI/II-Digits? y    ASAI Routing? y
Variables? n  CINFO? y      BSR? y      Holidays? n
01 wait-time  2      secs hearing ringback
02 goto       step 5      if available-agents in skill 3      < 1
03 queue-to   skill 3      pri m
04 stop
05 converse-on  skill 2      pri h passing vdn      and wait
06 collect    1      digits after announcement none
07 goto       step 13     if digits = 0
08 goto       step 12     if digits = 1
09 goto       step 11     if digits = 2
10 wait-time  10     secs hearing silence
11 wait-time  5      secs hearing silence
12 wait-time  5      secs hearing silence
13 disconnect after announcement none
14 stop
15
16
17
18
19
20
21
22

```

4. Configuration of Avaya Interactive Response

4.1. Installation of the CBM package on the AVAYA Interactive Response

Copy the Servion Callback Manager AVAYA Interactive Response package (SgslCBM.ds) to an installation directory (in this example /) on the AVAYA Interactive Response system and execute the following command to install the package:

```
pkgadd -d /SgslCBM.ds SgslCBM
```

```
sunblade(root)# pkgadd -d /SgslCBM.ds SgslCBM
```

The Servion Callback Manager application will be installed on the AVAYA Interactive Response system with the application files stored in the following directories:

Application File	Description
/vs/trans	Holds all the application executables
/speech/talk	Holds the phrase list files
/tmp	Temporary files like the xml responses
/voice1/vfs/talkfiles/220	Application talk files
/etc/conf/init.d	Shell for assigning the outbound application
/opt/cbm	Configuration files for IP

4.2. Crontab entry to check status of CBM application

Edit the crontab for adding a cron job using the command “*crontab -e root*” from the command prompt and add the following lines to the end of the file

```
#crontab entry to check if cbm is running
0,5,10,15,20,25,30,35,40,45,50,55 * * * * /opt/cbm/checkobm
```

The above entry shall keep checking whether the CBM is running or not once in every 5 minutes.

5. Configuration of Servion Call Back Manager

5.1. AVAYA Interactive Response

5.1.1. Configuration Parameters

Application related configurable parameters can be added using the following command:

```
/opt/cbm/reconfcbm
```

The configurable parameters are given below:

1. Site Name. The maximum length allowed is 7 alphanumeric characters.
2. Incoming VDN/Code. This is the Conversed Digit code the application will receive. Up to 5 numeric digits can be entered.
3. Feature Access Code. The allowed values are 1 or 2 numeric digits.
4. The Return Value for Success. The allowed values are 1 or 2 numeric digits.
5. The Return Value for Failure. The allowed values are 1 or 2 numeric digits.
6. The Return Value for VoiceMail. The allowed values are 1 or 2 numeric digits.
7. Network address (IP) of the CBM server. This parameter must be entered in a dotted quad IPv4 format (192.45.120.21 in the test configuration, or 172.42.173.53 in the following screen capture).
8. The Outdialing Code. The allowed values are either 1 or 2 numeric digits, or the string “NA” if no Outdialing code is required.

```
Enter The Site Name :SERVION
Enter The Incoming VDN/Code:277
Enter The Feature Access Code:73
Enter The Return Value For Success:0
Enter The Return Value For Failure:1
Enter The Return Value For VoiceMail:2
Enter The CBM Server IP:172.42.173.53
Enter The OutDialing Code eg(9 or 0 or NA):9
Shall I Commit The Configuration's (y/n) ? : y
sunblade(root)#
```

One or more sites can be configured on a single Avaya Interactive Response as long as they each have unique Incoming VDN/Codes. A configuration for a particular site can be removed by executing the **removeconf** command with the unique Incoming VDN/Code for the site to be removed as an argument. In the example below, the site configured with the Incoming VDN/Code of 1151 will be removed.

```
sunblade(root)# removeconf 1151
Removing The Configuration For : 1151
sunblade(root)#
```

5.1.2. Assigning the Servion Callback Manager Service to AVAYA Interactive Response Channels

To assign channels execute the command:

```
assign serv cbm_ibm startup cbm_ibm to chan <chan num-range>
```

In the example below the service is assigned to channels 3 through 5 inclusive.

```
sunblade(root)# assign serv cbm_ibm startup cbm_ibm to chan 3-5
Assigned service cbm_ibm startup cbm_ibm to channels 3-5
sunblade(root)# _
```

To reconfigure the channels assigned to Callback Manager, the above **assign** command can be used in conjunction with following **delete** command:

```
delete serv cbm_ibm from chan <chan num-range>
```

 **Note:**

The script for the outbound application is loaded into the inittab during the installation process, hence, after configuring the IP address of the CBM server, execute the command:

```
mkitab
```

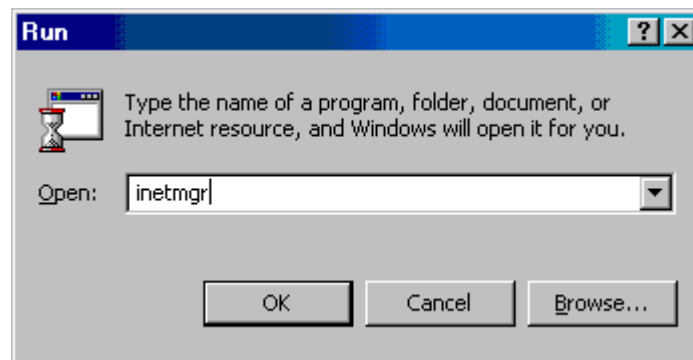
This command will rebuild the inittab and register the DIPs.

5.2. Windows Server

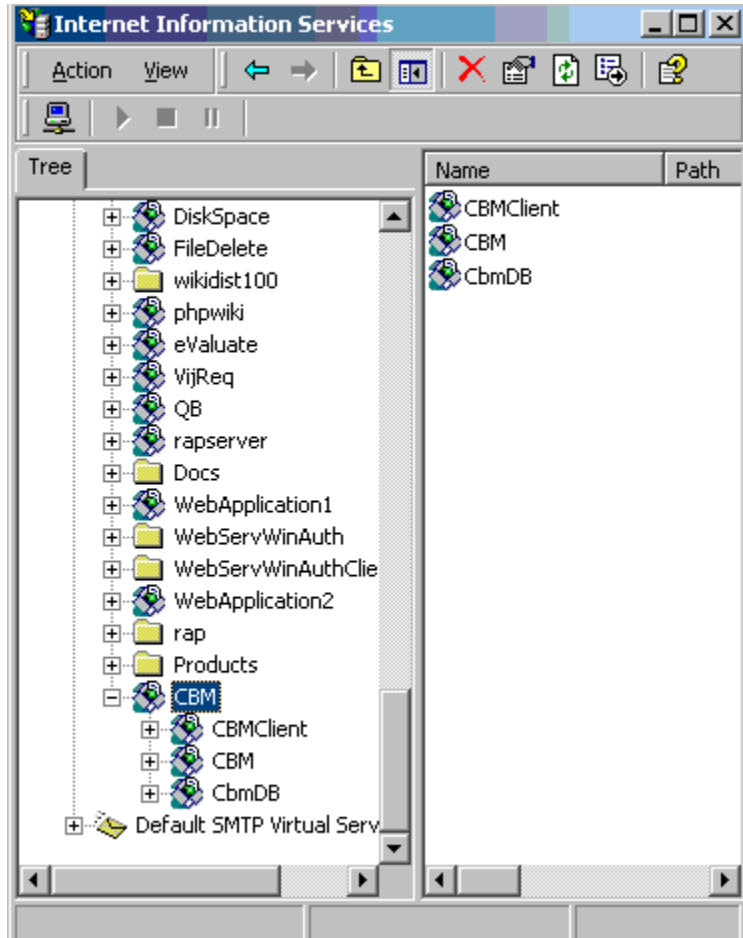
5.2.1. Web Services Application.

In order for users to have access to reports and administration features via the web interface, the **CBM** directory along with each of the folders inside this directory; **CBM**, **CBMClient**, and **cbmDB**, must first be converted into web folders. The Internet Information Services (IIS) Manager is used for this step.

Start the IIS Manager by clicking on **Start** and then **Run**. This will open the run window. Enter **inetmgr** and click the **OK** button to open the IIS Manager window.

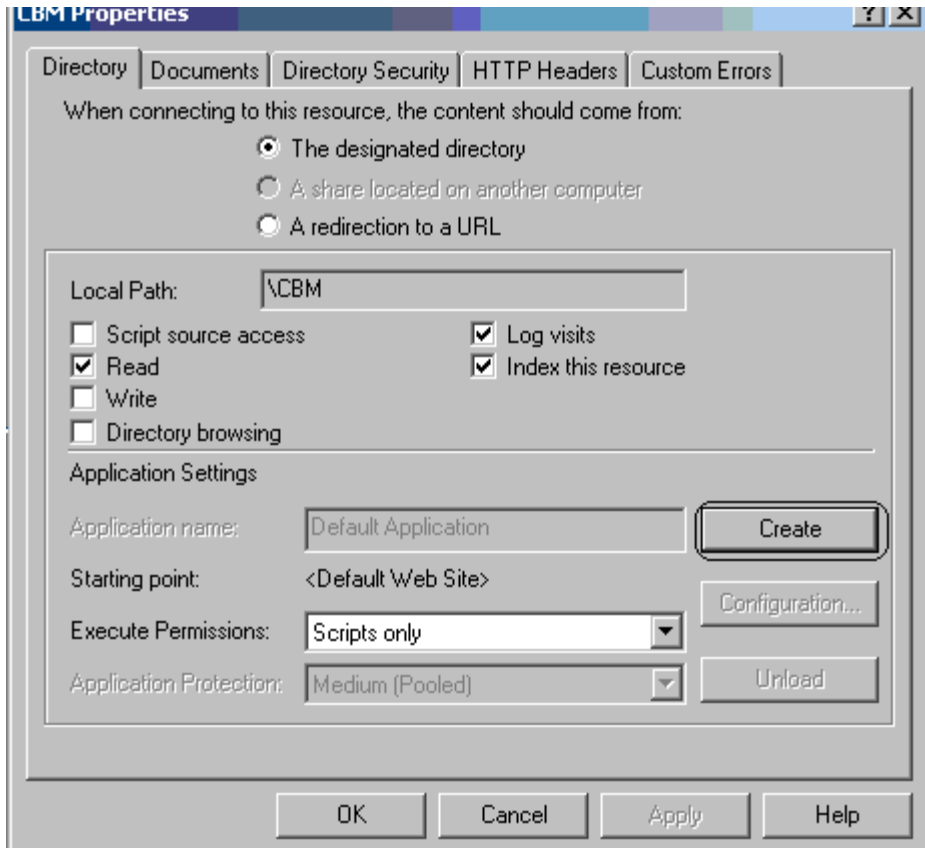


In the left hand file window, browse to and select the **CBM** folder. Expand the listing to show the three subfolders under this **CBM** folder. Right click the **CBM** icon and select the “**Properties**” option.

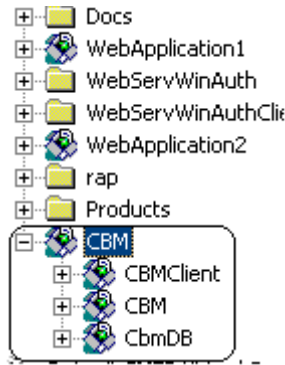


Click the “**Create**” button to create the virtual folder.

The following CBM Properties window will be displayed. Click the “**OK**” button



This completes the process for the **CBM** directory itself. The same process must be repeated for each of the three sub directories in the **CBM** directory. Complete the same steps as given above for creating virtual folders (Right click on one sub folder icon, select properties option, and click create button) for each of the three subfolders of CBM (Refer to the screenshot below for the sub folder names).



5.2.2. Web.config file

For the CBM application to function properly certain configurations have to be made in the web application.

Path: C:\inetpub\wwwroot\CBM\CBMClient\Web.Config

Name of the file: web.config xml file.

The web server IP needs to be configured in the web.config file for the web application to connect to the web server for calling the backend web services.

Given below is the extract of the web.config file for web server IP Configuration.

```

        <deny users="[comma separated list of users]"
        roles="[comma separated list of roles]"/>
    -->
    </authorization>
    <!-- APPLICATION-LEVEL TRACE LOGGING
    Application-level tracing enables trace log output for every page within an application.
    Set trace enabled="true" to enable application trace logging. If pageoutput="true", the
    trace information will be displayed at the bottom of each page. Otherwise, you can view the
    application trace log by browsing the "trace.axd" page from your web application
    root.
    -->
    <trace enabled="false" requestLimit="10" pageOutput="false" traceMode="sortByTime" localOnly="true" />
    <!-- SESSION STATE SETTINGS
    By default ASP.NET uses cookies to identify which requests belong to a particular session.
    If cookies are not available, a session can be tracked by adding a session identifier to the URL.
    To disable cookies, set sessionState cookieless="true".
    -->
    <sessionState mode="InProc" stateConnectionString="tcpip=127.0.0.1:42424" sqlConnectionString="data
    source=127.0.0.1;Trusted_Connection=yes"
    cookieless="false" timeout="20" />
    <!-- GLOBALIZATION
    This section sets the globalization settings of the application.
    -->
    <globalization requestEncoding="utf-8" responseEncoding="utf-8" />
    </system.web>
    <appSettings>
    <add key="webserviceIP" value="10.201.64.26" />
    </appSettings>
</configuration>

```

The default configuration comes with a default webservice IP address found between the **appSettings** tags. Replace this default IP address with the IP address of the Web server as circled below and save the file. In the test configuration, the webserver IP address is 192.45.120.21.

```
<appSettings>
  <add Key="WebserviceIP" Value="192.45.120.21" />
</appSettings>
```

Path: C:\inetpub\wwwroot\CBM\CBMDB\Web.Config
Name of the file: web.config xml file.

```
<appSettings>
  <add key="ConnectionString" value="Data Source=RES;User
ID=system;Password=manager;Persist Security Info=False;" />
  <add key="LogFilePath"
value="C:\Inetpub\wwwroot\PSICBM\CbmDB\logs\CBM.txt" />
  <add key="LogLevel" value="2"/>      <!--(0-Error, 1-Debug, 2-Info ) -->
  <add key="LogSize" value="2097152"/> <!--(in bytes:2 Mb file) -->
  <add key="MaxFiles" value="5"/>      <!--(No. of files to backup) -->
</appSettings>
```

5.2.3. Machine.config file

Open the file C:\WINNT\Microsoft.NET\Framework\v1.1.4322\CONFIG\machine.config and locate the two entries shown below:

```
<!-- <add name="HttpPost" /> -->
<!-- <add name="HttpGet" /> -->
```

Uncomment the above entries for the AVAYA Interactive Response application to communicate with the web server through the **HttpPost** and **HttpGet** as shown below:

```
<add name="HttpPost" />
<add name="HttpGet" />
```

5.3. Synchronize system clocks on Avaya Interactive Response and Windows Server

It is important that the system clocks on both the Avaya Interactive Response and the Windows-based Call Back Manager are synchronized. Call Back Manager allows for a few seconds in clock skew, however if the clocks diverge by more than this, problems can occur. Requests for call backs are stored based on the clock on Avaya Interactive Response. Call backs are made based on the clock on the Windows server. The system clock can be adjusted on Avaya Interactive Response using the **date** command (using the **root** login). Use the **man date** command for the usage. The system clock on the Windows PC can be adjusted by double clicking on the time in the lower right hand corner of the desktop.

6. Verification Steps

6.1. Software Installation Status

To check if the software package has been properly installed on Avaya Interactive Response, execute the command:

```
pkginfo -l Sgs1CBM
```

The status field should show **completely installed**, as circled below.

```
sunblade(root)# pkginfo -l Sgs1CBM
  PKGINST:  Sgs1CBM
   NAME:   Call Back Manager
  CATEGORY: Application
   ARCH:   i486
  VERSION: 1.1
   VENDOR: Servion Global Solutions Limited
  INSTDATE: Sep 01 2004 10:42
   STATUS:  completely installed
   FILES:  116 installed pathnames
           8 shared pathnames
           14 directories
           9 executables
           4271 blocks used (approx)

sunblade(root)#
```

6.2. Crontab Entry

The crontab on Avaya Interactive Response must contain an entry that checks the status of the Servion Callback Outbound Manager every 5 minutes. To check on this entry, use the following command:

```
crontab -l
```

Verify that the entry circled below is in the crontab listing.

```

# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0 /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#The following 2 Cron entries are for hourly and weekly jobs for print logs and
queue fix
10 * * * * /mtce/bin/prlrem
1 5 * * 6 /mtce/bin/prqrem
15 0 * * * /us/bin/util/cronsysmon
1 * * * * /us/bin/util/ckCoreFiles
15 * * * * /us/bin/util/rclean
15 0 * * * /us/bin/util/mufiles
15 0 * * * /us/bin/util/croncdh > /dev/null 2>&1
#The following entry will cleanup old backup files
15 2 * * * /us/bin/util/delbackup -s
# faxqcl crontab entry
15 1 * * * /us/bin/urs/faxqcleanup >/dev/null
#crontab entry to check if cbm is running
0,5,10,15,20,25,30,35,40,45,50,55 * * * * /opt/cbm/checkobm
sunblade(root)# █

```

6.3. AVAYA Interactive Response DIPs

To verify the necessary DIPs are running on Avaya Interactive Response, execute the following command:

```
/opt/cbm/checkdips
```

Verify that the each of the processes circled below are listed.

SLT	BBNAME	QKY	PID	INS	D	CDATE mm/dd, hh:mm:ss	WK	SKY	QID	RE- SPA	WKCNT
83	vntxDip	67	7975	0	Y	10/08, 06:32:21	R	83	0	0	18
84	vntxDip	67	7977	1	Y	10/08, 06:32:21	R	84	0	0	20
85	vntxDip	67	7979	2	Y	10/08, 06:32:21	R	85	0	0	32
86	vntxDip	67	7981	3	Y	10/08, 06:32:21	R	86	0	0	13
87	vntxDip	67	7983	4	Y	10/08, 06:32:21	R	87	0	0	20
113	CallPrgDip	97	8182	1	Y	10/08, 06:32:47	R	113	0	0	3
115	soft_srz	99	0	14	Y	10/08, 07:05:52	R	115	0	0	1
107	CBMDIP	91	8177	1	Y	10/08, 06:32:44	R	107	0	0	1

sunblade(root)# █

6.4. Vonetix Data Plugin

Vonetix is a required feature for Callback manager. (Note that although Vonetix requires a functional DNS for all of its functions to work, the subset of Vonetix functions that Callback manager requires do not need a working DNS.)

To check the status of Vonetix, execute the following command:

```
vonetix -status
```

The output of this command will look as follows if Vonetix is not running:

```
sunblade<root># vonetix -status
DNS is not configured on this server. Some functions of Vonetix
require a properly configured DNS and will not work without it.
Please contact Avaya for support.
Vonetix is not running
No Vonetix DIP processes are running.
sunblade<root>#
```

If Vonetix is running, the output will include the process table for the Vonetix DIPs, as shown below:

```
Vonetix DIPs are running:
NAME-INSTANCE    TIME RUNNING    PROCESSOR    TIME    ID
  untxDip-0      1-03:48:10      0:00      7957
  untxDip-3      1-03:48:09      0:00      7963
  untxDip-2      1-03:48:10      0:00      7961
  untxDip-1      1-03:48:10      0:00      7959
  untxDip-4      1-03:48:09      0:00      7965
```

If Vonetix is not running, it can be started by executing the command:

```
vonetix -start
```

```
*** DEFAULT SERVICES ARE NOT AVAILABLE
Installing inittab entries
Rebuilding inittab
Adding Web Administration link
Vonetix is starting
Waiting for Vonetix to come up...
Status:      running
Start Time: 2004-09-10 11:51:16.095
```

After starting Vonetix, re-check its status using the following command:

vonetix -status

```
Status:      running
Start Time: 2004-09-10 11:19:24.647
Vonetix DIPs are running:
NAME-INSTANCE  TIME RUNNING  PROCESSOR TIME  ID
  untxDip-0           25:42      0:00  1019
  untxDip-3           25:42      0:00  1025
  untxDip-1           25:42      0:00  1021
  untxDip-4           25:41      0:00  1027
  untxDip-2           25:42      0:00  1023
```

6.5. Servion Callback Manager

The status of both the Inbound Manager and Outbound Manager can be checked with the following command:

sysmon

System Monitor - Voice Channels						
Channel	Calls Today	Voice Service	Service Status	Caller Input	Dialed Digits	
0	9		*On Hook			
1	0		*On Hook			
2	0		*On Hook			
3	6		*On Hook			Inbound
4	6	cbm_ihm	Collect			
5	13		*On Hook			
6	0	cbm_obm	Suspended		5204	
7	0	cbm_obm	vntxDip		5204	Outbound
8	0	cbm_obm	vntxDip		5204	
9	0		*On Hook			
10	22		*On Hook			
11	0		*On Hook			

7. Conclusion

Interoperability Compliance Testing concluded that Servion Call Back Manager version 1.1 successfully interoperates with Avaya Interactive Response version 1.2.1. This was confirmed using both the Windows 2000 Server and Windows 2003 Server operating systems. It is important to note that testing determined that the installation of service pack 1.0 for Microsoft .NET Framework is a critical requirement for reliable operation of Servion Call Back Manager.

8. Additional References

The following documents can be found at <http://support.avaya.com>:

1. Administrator's Guide for Avaya Communication Manager, Issue 8, June 2004; Doc ID: 555-233-506
2. Avaya Communication Manager Call Center Software Call Vectoring and Expert Agent Selection (EAS) Guide, Issue 1, June 2004; Doc ID: 07-300186

©2005 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 Developer*Connection* Program at devconnect@avaya.com.