



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

SIP Voicemail HA Deployment Guide

Known Issues and Recommendations

12/17/2025

Contents

- 1 Known Issues and Recommendations
 - 1.1 Primary Server Disconnection
 - 1.2 Subnet for GSVM Servers

Known Issues and Recommendations

Primary Server Disconnection

When the primary server goes down due to power or network disconnection, the script to disable the Virtual IP fails because the primary box is unavailable. When the primary server is started, there might be an IP conflict since Virtual IP is not disabled in the primary server. Genesys recommends you add the disable Virtual IP script as a startup script in the machine, so the Virtual IP is disabled by default when a system boots up and then according to the event (4563) from Genesys application in our case SIP Server the reaction scripts should be triggered. See **Creating Virtual IP control scripts** ([Windows](#), [Linux](#)). **Note:** There is no resolution for an IP conflict caused by unplugging an IP cable. This is a known limitation.

VIP Not Enabled Issue

The VIP not enabled issue can only occur in Windows. In this scenario, the backup server is down when the primary server starts. If the VIP down service is started after the Primary SIP server is started and the alarm condition from the server has already enabled the VIP in the machine, the VIP is disabled and VM SIP server goes into a service unavailable state.

Preventing The VIP Not Enabled Issue



Purpose: To prevent the VIP from being disabled and the VM SIP server from going into a service unavailable state.

Start

1. Add the following script as a startup script in Windows.

[+] Commands for SC_ENABLE_VIP.BAT

```
@echo off
rem Set Application, Configuration and SCS server details
rem Application Details
set application=<Name of the Application>
rem Configuration server details
set host=<IP address of the config server>
set port=<Port of the config server>
set appname=<App name the config server>
set user=<Config server user name>
set password=<Config server password>
rem Third party application details
set thirdparty=<Third party server name that is linked to the VIP_ON script>
rem for example thirdparty=primary_VIP_up
rem SCS server details
set scshost=<IP address of the SCS server>
set scsport=<Port of the SCS server>
rem set the path in which log files will be stored
```

```
Logpath= <Specify the path to log files>
rem for example Logpath= C:\Solution_HA\Logs
rem set the mlcmd utility path of the local machine in which it is copied
mlcmdpath= <Specify the mlcmd path >
rem for example mlcmdpath = C:\Solution_HA\Mlcmd
echo %date% >> %Logpath%\thirdparty_status.log
echo %time% >> %Logpath%\thirdparty_status.log
:start
echo checking the status of thridparty application >> %Logpath%\thirdparty_status.log
"%mlcmdpath%\mlcmd.exe" -getappstatus-runmode %thirdparty% -cshost %host%
-csport %port% -csappname %appname% -csuser %user% -cspassword %password%
-scshost %scshost% -scsport %scsport% 1> %Logpath%\application_mode.txt 2>&1
FINDSTR "APP_STATUS_STOPPED" %Logpath%\application_mode.txt > nul
if errorlevel 0 goto next
if not errorlevel 0 goto start
:next
rem Checking the mode of application
"%mlcmdpath%\mlcmd.exe" -getappstatus-runmode %application% -cshost %host%
-csport %port% -csappname %appname% -csuser %user% -cspassword %password%
-scshost %scshost% -scsport %scsport% 1> %Logpath%\application_mode.txt 2>&1
FINDSTR "BACKUP" %Logpath%\application_mode.txt > nul
if not errorlevel 1 goto backup
rem Checking the mode of application
"%mlcmdpath%\mlcmd.exe" -getappstatus-runmode %application% -cshost %host%
-csport %port% -csappname %appname% -csuser %user% -cspassword %password%
-scshost %scshost% -scsport %scsport% 1> %Logpath%\application_mode.txt 2>&1
FINDSTR "PRIMARY" %Logpath%\application_mode.txt > nul
if errorlevel 1 goto next
timeout 3
if errorlevel 0 goto end
:backup
echo %application% is not in Primary mode, exiting the loop
>> %Logpath%\thirdparty_status.log
exit
:end
rem stopping application
"%mlcmdpath%\mlcmd.exe" -stopapp %application% -cshost %host% -csport %port%
-csappname %appname% -csuser %user% -cspassword %password% -scshost %scshost%
-scsport %scsport%
timeout 3
rem starting application
"%mlcmdpath%\mlcmd.exe" -startapp %application% -cshost %host% -csport %port%
-csappname %appname% -csuser %user% -cspassword %password% -scshost %scshost%
-scsport %scsport%
timeout 3
echo %application% is restarted since the application %thirdparty% was in unknown
state >> %Logpath%\thirdparty_status.log
echo %date% >> %Logpath%\thirdparty_status.log
echo %time% >> %Logpath%\thirdparty_status.log
echo. >> %Logpath%\thirdparty_status.log
```

2. Copy the MLCMD utility files (mlcmd.exe and mlcmd.exe.manifest files) from the SCS host onto the host running the VM components.

End

Subnet for GSVM Servers

You must connect both the primary and backup GSVM Servers within the same subnet to prevent problems refreshing the ARP cache of the gateway after the switchover of the Virtual IP. You can use the ARPING command in the Virtual IP takeover scripts to resolve this issue. ARPING is a tool included in Linux. A third-party version of ARPING can also be downloaded and installed in Windows.