Reporting and Monitoring

This chapter discusses various types of reporting and monitoring functions that can be used with CVP, and covers the following areas:

- Real Time Health Monitoring of CVP Components, page 16-1
- Statistical Monitoring of CVP Components, page 16-2
- End to End Tracking of Individual Calls, page 16-2
- Formal Reporting Based on ICM Data, page 16-3
- Formal Reporting Based on CVP VoiceXML Server Data, page 16-3

Real Time Health Monitoring of CVP Components

There are several ways to monitor the health of CVP components. CVP VoiceXML Server does not have any built-in health monitoring capabilities other than the SNMP facilities offered by the operating system.

The CVP Call Server continues to use the Cisco Standalone Distributed Diagnostics and Services Network (SDDSN) to provide alarm reporting through a variety of mechanisms such as:

- SNMP traps
- CiscoWorks2000 Syslog, which receives log messages and permits queries on the logs
- Microsoft Windows Remote Access Server (RAS) and Hosted IPCC or ICM Event Management System (Cisco Remote Monitoring Suite, traditionally called Cisco Phone Home)

The ICM Remote Monitoring Suite's AlarmTracker is often used as the mechanism to monitor system status for multiple CVP machines. The CVP Call Server sends alarms to SDDSN, and you can use the Alarm Tracker to view alarm status across multiple CVP machines.

For detailed information about the CVP Call Server’s entire SNMP implementation, refer to the *CVP 3.1 Configuration and Administration Guide*, Chapter 7, “Alarm Handling and Logging.”

Cisco gateways provide for SNMP monitoring. Information about various IOS MIBs are publicly available at Cisco.com; of particular relevance is the ISDN MIB, information about which can be found at:

Statistical Monitoring of CVP Components

This section focuses on ways to monitor the number of calls being handled by each component or by the contact center as a whole, either real-time or historically.

While no single tool provides system-wide monitoring, there are a number of tools available for monitoring system components:

- Gateway statistics, including historical and real-time call counts, may be obtained via IOS commands.
- The gateway’s ISDN MIB referenced above can be used to provide overall call volumes to an SNMP console, though it does not help distinguish which calls are in self service, in queue, or talking to agents.
- The ICM provides reports on the number of VRU ports in use at half hour intervals and present this information in discreet or aggregated form. If you know how many ports are available, you can determine whether there were all-trunks-busy periods, and for how long.
- The CVP Call Server stores detailed statistics in its log files every half hour (a configurable interval) indicating how many calls arrived, were transferred, encountered errors, etc., as well as how long they lasted and many other specific pieces of information. These are described in the CVP 3.1 Configuration and Administration Guide in Chapter 5, Engine Administration, and in Chapter 7, Viewing Voice Browser Logs. These statistics are however not in a form which is suitable for loading into a database or spreadsheet.

This document is available at:


- The CVP VoiceXML Server writes data rows to a flat log file every for every new call it handles. This data is in comma-separated-value format and quite suitable for loading into a database.

End to End Tracking of Individual Calls

When a call arrives at a CVP ingress gateway, IOS assigns that call an H.323 Conference ID, which is a 36-digit hexadecimal GUID that uniquely identifies the call. CVP carries that GUID through all of the components that the call encounters, as shown below:

- Ingress gateway – shown in IOS log files
- VXML gateway – shown in IOS log files
- CVP Call Server – shown in CVP Voice Browser and CVP Application Server log files
- CVP VoiceXML Server – shown in detailed call logs, and available to VoiceXML Server applications for storage or for passing on to further back-end systems
- ICM – shown in the ECC variable user.task.id and stored with all TCD and RCD records
- ASR and TTS servers – shown in logs as the logging tag
- Cisco CallManager – appears in the detailed logs

Thus, with proper levels of logging enabled, a call can be traced through all of the above components.
Formal Reporting Based on ICM Data

ICM 6.0(0) introduced a series of reporting enhancements, among which was a new facility for VRU Progress Reporting. This facility made it possible to identify segments of a VRU application, and then keep track of how many calls entered and left those segments, how many successfully completed their transactions, and how many transferred out or abandoned intentionally or due to errors. VRU Progress Reporting is designed for use with complex self service applications that use CVP Microapplications – a practice which has since fallen into disuse in favor of the CVP VoiceXML Server. However, it is still possible to make good use of this facility by dividing VoiceXML Server applications into segments or transactions, and using ICM scripting to handle the top level self service menu. This way, the ICM can still keep and present formal statistics on the success and failure rates of each application segment.

Formal Reporting Based on CVP VoiceXML Server Data

As mentioned above, CVP VoiceXML Server provides a comma-separated-value file in which information about each call is logged. In fact, if you turn debug settings high enough, you receive a large amount of logging, including the results of each node in each application for each call. This is a large volume of data, but for some purposes you require that level of detail. For example, you may be interested in capturing a great deal of call data for the first few weeks of an application’s production use, or in order to troubleshoot a particularly difficult intermittent problem.

Cisco generally does not recommend that this level of call logging be turned on for long periods of time, due to throughput concerns. However, if you are sure you are under-utilizing your CVP VoiceXML Servers, this is a method of capturing detailed application information. Be aware however, that though VoiceXML Server rolls over to new log files every day at midnight, it does not automatically purge old log files. This amount of data can quickly saturate the disk if you do not put operational measures in place to ensure that does not happen.

With custom report development, you can process the CVP VoiceXML Server log files to produce the following types of reports, among others:

- Application reporting: Call counts and call duration summaries per application
  - Source: Application log files under each application folder
  - Levels: All levels of reporting

- Application reporting: Call source summary data such as call counts by ANI
  - Source: Application log files under each application folder
  - Levels: All levels of reporting

- Application menu usage call counts and durations: This data can provide some insight into how a caller transverses through the menu setup for the self-service application.
  Important: There could be system performance degradation with complete logging.
  - Source: Application log files under each application folder
  - Levels: Moderate and complete logging only

- Voice elements reporting: This data can provide insight into interaction with voice elements including basic speech statistics. Leverage application log files under each application folder.
  - Source: Application log files under each application folder
  - Levels: Moderate and complete logging only
• Call detail reporting: This data can provide flow details about every call made to the system and can help trace a particular call through the self-service application.
  – Source: Log files under each root folder
  – Levels: All levels of reporting