VoiceGenie 7.2

Tools

User's Guide

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library CD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-118-974-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-5649-6871	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the Genesys Technical Support Guide for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the Genesys 7 Licensing Guide.

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 09-2007

Table of Contents

Chapter

1 Introduction

1.1 Overview

The VoiceGenie Platform is accompanied by a set of tools that provide insight into the performance, usability and quality of the applications it executes. The VoiceGenie Call Analyst and VoiceGenie Quality Advisor tools collect data from all of the calls handled by your deployment and provide convenient interfaces for navigating and analyzing the information. This user's guide provides detailed instructions on using these tools to monitor and optimize your deployment.

1.1.1 Call Analyst

The VoiceGenie Call Analyst is a multifaceted tool providing insight into the performance and usability of applications running on the VoiceGenie platform. The key functional aims of the product are to:

- Provide information on the frequency with which specific applications and features are being used.
- Highlight caller behavior one call at a time and to provide overall statistics to uncover trends across multiple calls.
- Provide useful metrics to help gauge the overall user experience offered by specific applications.
- The resulting benefits include the facilities to:
- Assist developers and support teams in troubleshooting applications.
- Identify potential problems with call flow and prompts.
- Better allocate development, platform and tuning resources to correspond with usage trends.

The Call Analyst includes a graphical desktop client for viewing statistics and logs generated by the VoiceGenie platform. The client is only a component of the Call Analyst; the remaining portions of this tool are composed of run-time

servers which gather and store data from a VoiceGenie deployment, and serve the data to the client when needed. This document provides background information and describes how to install and use the Call Analyst.

The Call Analyst client is implemented as a plug-in for the Eclipse platform (see `http://www.eclipse.org`), and can be used along side any other Eclipse tool or plug-in. As such, some familiarity with Eclipse is recommended before reading this document in order to provide some basic context. Understanding how to use Eclipse for Java development is not necessary; however, it would be beneficial to have some understanding of basic Eclipse concepts such as the *workbench, perspectives, views, editors, resources*, and *projects.*

1.1.2 Quality Advisor

The VoiceGenie Quality Advisor provides statistics on the service quality of deployments of the VoiceGenie platform. Service quality is a measurement of the quality of a speech application deployment from the user's perspective. This differentiates the analysis from operations logging and reporting which typically focus on the availability and performance of servers and system components. For example, the failure of individual servers may not impact a service in a way that is perceivable by users if failover mechanisms work as designed, and as such, these events should not impact service quality measurements. These events would, however, impact other kinds of operational reports and analyses. On the other hand, extreme system load may impact a deployment in a way that slows down an application's responsiveness but does not cause any requests or transactions to time out. This event would impact service quality, but might not show up in any operational logs.

Service Quality is measured in a way that takes all calls to the system into account. Thus, rather than using a "sampling" methodology where periodic test calls are made to the system, the platform itself takes measurements of the calls it is handling. The product will gather information from instrumentation of the platform and the applications running on the platform and prepare reports on quality at regular intervals. Alarms can be generated if quality falls below an acceptable level. Supplemental information such as histograms of the latencies involved in external interactions accompany the quality report to provide the necessary background information to address any problems.

1.2 Feature Summary

1.2.1 Call Analyst

The Call Analyst provides two main sets of features:

- Generation of various statistical reports providing insight into overall application performance and usability; and
- Collection and display of detailed call logs to allow individual calls to be analyzed.

The statistical reports are based on *all* calls that are handled by the VoiceGenie deployment. The reports are organized into two groups: operational reports and usability reports. The following operational reports are provided:

- **Call Distribution** – this report provides a histogram describing the arrival pattern of calls to a platform (or cluster) over a specific period of time.
- **Call Length Distribution** – this report provides a histogram describing the length of calls to a platform (or cluster) over a specific period of time.
- **System and Application Loading** – this report provides a histogram describing the maximum number of concurrent calls to a platform (or cluster) in discrete intervals over a specific period of time

Additionally, the following usability reports are provided:

- **Recognition Performance Summary** – This report provides recognition performance statistics that give a high level analysis of how effective each part of the application is at gathering input. This essentially captures the number of times the user must typically be re-prompted at each point in the application.
- **Recognition Result Frequency** – This analysis provides statistics on the usage of different input tokens in voice and dtmf grammars. For each point in the application, the analysis provides a table containing the individual voice and dtmf input tokens used in that context over the specified time period.
- **Collection Attempt Details** – This analysis provides statistics on the attempts to collect input at each interaction in the application. For each recognition context a table is provided which details the outcome of the collection attempts executed at that context. The collection attempts are organized by the number of re-prompts in a visit to a recognition context.
- **Menu Selection Frequency** – This analysis shows the frequency with which the various choices in a menu dialog are selected.
- **Barge-In Analysis** – This report provides information the percentage of users who barge-in over prompts, and the amount of time they wait before barging in.

- **Transfer Analysis** – This analysis provides statistics on the outcome of transfer execution.

Additionally, statistics can be saved and loaded from disk (using an XML format), or exported to CSV, or HTML.

Analysis of usability and performance statistics may provide insight into problematic areas of a voice application. However, detailed traces of individual calls are often required to properly assess the cause and eventual solution of usability problems.

Call analyst can be used to collect detailed logs of individual calls. However, since VoiceGenie platform deployments often handle thousands of calls each day, collecting logs for all calls may result in much more data than could ever be analyzed. For this reason, the Call Analyst allows a user to subscribe to a specified percentage of calls to an application. The user can collect their subscribed call logs using the Call Analyst client.

The call logs are stored on the client machine in an XML format. When viewed in the Call Analyst client, they are displayed in a convenient, collapsible tree of events. Call logs stored on a client machine can be searched by ANI, DNIS, Call time and application name.

Statistics and call logs can be organized into *workbook* directories on the client machine's disk drive. Workbook data can be transferred to another machine and accessed even if there is no network access to the VoiceGenie deployment.

1.2.2 Quality Advisor

The service quality advisor tool measures the service quality and performance of applications running on VoiceGenie deployments. Service quality is a measurement of the quality of a speech application from the users' perspective and is expressed as a percentage of calls that meet acceptable quality criteria. At a high level, the criteria are as follows:

- The call is answered within an acceptable amount of time.
- User perceived latency is below an acceptable threshold.
- There is no noticeable gap in audio playback.
- No application failures occur (indicated by the application via log tags).
- The call does not disconnect prematurely due to a platform error.
- All calls are monitored and flagged if they do not meet the quality criteria. Hourly, Daily, Weekly, and Monthly reports are generated to indicate the percentage of calls that meet the quality criteria.

In order to allow operations teams to react to sudden drops in service quality, the percentage of calls passing the criteria are calculated at frequent intervals and alarms are generated via the VoiceGenie OA&M Framework if quality drops below a threshold.

Latency histograms for call answer time and user perceived latency will be available for analyzing the service outages. These histograms will be available for the same time periods covered by the service quality report. Furthermore, histograms will also be available for all system operations and interactions that could contribute to the user perceived latencies. These histograms include but are not limited to:

- VoiceXML page fetch time
- VoiceXML page compile time
- JavaScript execution
- Audio prompt fetch time
- TTS server response time
- MRCP message response time
- Call setup time (incoming and outgoing calls)

For each measured latency, users can configure acceptable values for a given percentile; for example, the user may state that calls must be answered within 2 seconds 95% of the time. The performance of the system against these criteria accompanies every service quality report. Errors will be logged if these criteria are not met.

The primary interface for accessing service quality and latency statistics is a web interface.

Chapter

2

Terminology and Basic Concepts

2.1 General (applies to all tools)

2.1.1 Analysis Target

A single VoiceGenie deployment may host several applications at once. When a call arrives at a platform the application that is executed is usually determined by the DNIS (the called party's number or URI). Alternatively, the application may be selected based on some other criteria. For example, the user may select the application from a menu. Applications often need to be considered separately during analysis; for example, a user may wish to only see recognition results for a specific application. The Call Analyst provides a facility to define distinct *Analysis Targets* that are used to group calls and statistics.

An analysis target is defined by a name. Every call that a deployment handles is assigned to an Analysis Target. Analysis Targets can be identified in two ways:

- An Analysis Target name can be associated with a specific DNIS. In the DNIS-URL mapping configuration page of the SMC, a parameter called `APPLICATION` can be added to a DNIS. The value of this parameter will be used as the Analysis Target name of any calls made to that DNIS.

- The VoiceXML `<meta>` tag can also be used to assign a call to an analysis target. This is done by setting the `application` meta data property for a VoiceXML page to the name of the analysis target. For example, including `<meta name="application" content="My Analysis Target"/>` in a VoiceXML page assigns a call that executes that page to the Analysis Target named `My Analysis Target`.

A call can only be assigned to a single Analysis Target and it cannot be reassigned. Thus, if a call executes two pages that set the application meta data property to different values, the call will be assigned to the value that is specified first. Also, the analysis target specified by the DNIS-URL mapping configuration takes precedence over any target specified in the VoiceXML pages executed by the call.

Calls that cannot be mapped to an Analysis Target based on DNIS or meta data are assigned to a special target called `UNKNOWN`. It is good practice to avoid assigning calls to the `UNKNOWN` analysis target, even if only a single application is handled by the platform as this complicates the organization of data if new applications are added in the future.

All statistics are calculated by the Call Analyst and Quality Advisor for individual Analysis Targets. Statistics can also be reported for all calls handled by the platform, regardless of analysis target – this is essentially an aggregation of the statistics for all analysis targets handled by the platform. A Call Analyst user may also subscribe to an analysis target so that the logs for some percentage of calls assigned to that target are collected for the user. This allows a sampling of calls to be gathered for usability analysis of a specific application.

Access to an analysis target's statistics and call logs can be controlled through user permissions. Users and analysis targets are defined through the VoiceGenie Tools Web Console. This tool is also used to grant users permission to access specific analysis targets. This prevents call analyst users from accessing each other's applications in hosted environments.

2.1.2 Time Granularities

The data that Tools collects and reports is grouped into several time granularities: hourly, daily, weekly and monthly. When viewing and generating reports, the granularity along with a time range defines the data that is returned.

- `Hourly:` gets the data for each hour in the time range
- `Daily:` gets the data for each day in the time range. A day is defined as the 24-hour period between 12:00 am of one day and 12:00 am of the next day.
- `weekly:` gets the data for each week in the time range. A week is defined to start at 12:00 am on Monday, and end at 12:00 am of the next Monday.
- `Month:` gets the data for each month in the time range. A month is defined to start at 12:00 am on the 1st of the month, and end at 12:00 am on the 1st of the next month.

Note that when requesting reports, the time range is normalized to include full time periods of the requested granularity. For example, if a report is requested with granularity `weekly` and with the time range starting April 11th, 2006 and

ending April 18th, 2006, the returned results will actually include two weeks' worth of data: for the week starting April 10th, and the week starting April 17th.

2.2 Call Analyst

2.2.1 Call Log Subscription

A call log subscription is a request for logs for a specific analysis target to be gathered for a specific user. A user can create, edit or delete a subscription using the Call Analyst Eclipse Client. When a user creates a subscription the analysis target is specified along with a percentage of calls that should be gathered. Once the subscription is created the system will collect the specified percentage of calls that match the analysis target. For example, if a user creates a subscription with analysis target `Voicemail Application` and a percentage of 5%, then 1 in 20 calls to the Voicemail Application will be collected for the user.

2.2.2 Contexts

The Call Analyst uses the recognition contexts that are inherent in a VoiceXML application to organize and group the statistics that it calculates. There are three kinds of contexts: Recognition Contexts, Menu Contexts, and Transfer Contexts. Each context type identifies areas of an application where certain actions, such as a recognition, take place.

Recognition Contexts

A recognition context is a point in the application where voice and/or dtmf recognition can take place. For example, a VoiceXML menu is a recognition context; a specific field in a form is another recognition context. A recognition context is uniquely identified by the combination of a page URL, a dialog (menu or form) id, and an optional form item name. The form item name is only necessary when the dialog id refers to a form. For example, a VoiceXML page `http://myservice.com/page.vxml,` the id of a form on the page, say `form1,` and the name of a field in the form, for example, `cityField` together identify a recognition context.

Menu Contexts

Menu Contexts are the subset of Recognition Contexts that are VoiceXML Menus. A menu context is uniquely identified by the combination of a page URL and a menu id.

Transfer Contexts

Transfer Contexts are similar to Recognition Contexts, except that they identify points in an application where a transfer can take place. All transfer contexts occur at transfer form items and are uniquely identified by a page URL, a form id, and a transfer form item name.

Ensuring that Contexts can be uniquely identified

Because many applications use a host's IP in the URLs that it uses (e.g. `http://10.0.0.1/page.vxml` rather than `http://myservice.com/page.vxml`), it is often the case that a page of an application is specified by multiple URLs such as `http://10.0.0.1/page.vxml` and `http://10.0.0.2/page.vxml` etc. For this reason, the Call Analyst only looks at the `request-id` portion of a page when identifying a recognition context. For example, `/page.vxml` is used rather than `http://10.0.0.1/page.vxml.` Of course, this may cause problems when two separate applications use the same `request-ids` such as `http://app-one.com/page.vxml` and `http://app-two.com/page.vxml;` this is why properly identifying applications with analysis targets is so important. The analysis target that a page maps to serves as a way to distinguish it from an identically named page in another application.

It is also important to note that the query string portion of a URL is not used when identifying the current recognition context. For example, when executing a page with URL `http://myservice.com/servicename/page.jsp?variable=value&anothervariable=anothervalue,` only the string `/servicename/page.jsp` will be used to identify the current recognition context. This can cause problems in some dynamic VoiceXML applications where a single JSP file is used to generate pages that represent different parts of the application. In order to resolve this issue, the form and menu ids should be used to identify the part of the application that is being executed.

2.2.3 Collection Attempt

Many of the recognition performance statistics calculated by the Call Analyst are presented in terms of collection attempts. A collection attempt represents one execution of a recognition context. Some of the statistics number the collection attempt to denote the ith consecutive attempt to gather information for a recognition context. *Note that this count is reset every time that the recognition context is successful (e.g filled) or changes (e.g a goto is executed in a noinput handler).*

2.2.4 Workbook

A Call Analyst workbook is a type of Eclipse project for managing call logs and analysis reports produced by the Call Analyst Server. Each workbook is associated with a subdirectory of the Eclipse workspace. All files that are added to the workbook are copied into its directory. One Eclipse workspace can contain multiple workbooks; this makes the workbook a convenient mechanism for organizing logs and analysis results.

2.3 Quality Advisor

2.3.1 User Perceivable Failures

The Quality Advisor bases its quality measurements on the number of calls with failures of which the user is aware. These errors include user perceived latencies such as the time to answer a call, the time to play the first prompt in a call, or the time to play a prompt in response to user input. It also includes application failures that the user is notified of with a failure prompt, and system failures that result in an aborted call. It specifically does not include request failures or transaction failures that cannot be detected by the caller. The categories of user perceivable failures are as follows:

Call Control Latency Failures

- **Call Answer Latency** – This failure occurs when the time between a call being offered to the platform and the call being answered by the platform is greater than the configured threshold.
- **Call Reject Latency** – This failure occurs when the time between a call being offered to the platform and subsequently rejected by the platform is greater than the configured threshold.

User-Perceived Latency Failures

- **Inbound First Prompt Latency** – This failure occurs when the time between a call being answered and either a) the first prompt *or* b) the end of user input, is greater than the configured threshold.
- **Outbound First Prompt Latency** – This failure occurs when the time between the platform connecting a call to either a) the first prompt *or* b) the end of user input, is greater than the configured threshold.
- **Inter-Prompt Latency** – This failure occurs when the time between playback of consecutive prompt queues is greater than the configured threshold. This latency will ignore any fetch audio prompts that are played,

and is not applicable when caller interaction occurs between playback of consecutive prompt queues.

- **Cumulative Response Latency** – This failure occurs when any of the DTMF input, ASR input, No Input, Recording, or Transfer response latencies exceed their respective thresholds. The following is a description of the response latencies:
  - The **DTMF Input Response Latency** measures the time between the end of DTMF input and either the subsequent prompt or the next end of user input. The measurement ignores No Input events that occur between the DTMF input and subsequent prompt or end of user input.
  - The **ASR Input Response Latency** measures the time between the end of ASR input and subsequent prompt or end of user input, ignoring any No Input events that occur in between.
  - The **No-Input Response Latency** measures the time between the end of a No Input and subsequent prompt or end of user input, ignoring any No Input events that occur in between.
  - The **Recording Response Latency** measures the time between the end of a recording and subsequent prompt or end of user input, ignoring any No Input events that occur in.
  - The **Transfer Response Latency** measures the time between the end of a transfer and subsequent prompt or end of user input, ignoring any No Input events that occur in between.
- **Audio Gap** – This failure occurs when an audio gap detected during prompt playback is greater than the configured threshold. An audio gap occurs when audio data cannot be sent to the telephony hardware or across the network at the expected rate because of system load or latencies incurred by the audio source (when the audio is streaming). Note that audio gap will be logged during the playback of a video-only prompt.

Application Error

A caller typically experiences a perceivable application failure when an error prompt is played to the user. As the system is not able to distinguish error prompts from other prompts, the application must indicate the failure to the system. This can be done by logging an error with the VoiceXML `<log/>` tag using a label of `com.voicegenie.quality.failure`. For example:

```
<audio src="errormsg.wav">I'm sorry I can't process your transaction
   right now.</audio>
<log label="com.voicegenie.quality.failure"/>A database error
   occurred.</log>
```

The default error handlers in the default VoiceXML file (`defaults.vxml`) are already instrumented with logs that will indicate that a user perceived application failure occurred.

System Error

This type of failure occurs when a call terminates abnormally due to a system error or crash. A call is considered to have ended abnormally when:

1. a call ends with a `syserr` reason code (which indicates a system error)

2. an inbound call is rejected, and the reject reason is not one that has been configured to be an acceptable reason. The default configuration allows an inbound call to be rejected with reason `decline` without being considered a system error. (see 5.3.1 Service Quality Parameters)

3. an outbound call is rejected , and the reject reason is not one that has been configured to be an acceptable reason. The default configuration allows an outbound call to be rejected with reason `decline, busy, fax, no answer,` or `hangup` without being considered a system error. (see 5.3.1 Service Quality Parameters)

4. an inbound call has been offered but the system never reaches a state where the call has been answered or rejected

5. an outbound call has been initiated but the system never reaches a state where the call has been answered or rejected

6. an inbound call has been answered but the system never reaches a state where the call is ended

7. an outbound call has been connected but the system never reaches a state where the call is ended

2.3.2 Service Quality

Service Quality is a simple measure of the quality of a deployed application (or set of applications) over a given period of time. This figure is expressed as the percentage of calls that have not experienced a user perceivable failure (see above) within a specified period of time.

The service quality analysis measurement reported by the Quality Advisor provides a guaranteed *maximum* measurement of system quality. In other words, the actual perceived quality could be lower than what is reported but is never better. The Quality Advisor runs on the platform being analyzed and detects failures on all calls being handled by the system. While system timings are measured as accurately as possible, there may be latencies that cannot be measured or errors that cannot be detected. This is because errors and latencies in the network, the telephony hardware, and low level software such as the hardware drivers or kernel are not captured by the tool. Thus there may be calls that experience failures that are not reported by the tool.

Thus, the tool is used most effectively as part of a two-tiered approach to service quality analysis. The Quality Advisor provides a *white-box, complete* approach to measuring quality where measurements are taken by the system

itself for every call it handles. This can be supplemented with a *black-box, sample-based* approach where measurements are based on automated test calls that are periodically placed to the platform. This sample-based approach is not able to report failures on all calls handled by the system but is able to detect failures that the Quality Advisor might miss, such as network failures. Combining these approaches will result in a very comprehensive measurement of service quality.

2.3.3 Service Quality Period & Service Quality Threshold

The Quality Advisor measures Service Quality at regular intervals called the *Service Quality Period.* The Service Quality is calculated by measuring the percentage of all calls that *ended* during the service quality period that experience a user-perceivable failure. When Service Quality is calculated for a period, it is compared to a configurable *Service Quality Threshold.* If Service Quality falls below the threshold a critical error is logged to the OA&M infrastructure. An SNMP trap can be raised when this error is logged, allowing operations staff to be immediately notified. This mechanism allows the Service Quality measure to act as a warning system that detects lapses in quality.

It should be noted that if a trivial amount of calls were handled in a Service Quality period, one or two failed calls can have a serious impact on service quality. In order to prevent unnecessary alarms from being raised, the Quality Advisor can be configured so that a warning event is logged instead of a critical event in these cases. This minimum number of calls that must have ended in a service quality period for a critical event to be logged is controlled by the minimum calls parameter.

Even with this parameter, it is important to configure the length of the Service Quality Period so that a sufficient number of calls complete during each interval. Typically, you want to make sure that one or two failed calls don't drop Service Quality far below the threshold. For example, if your threshold is 98% you would want to make sure that the Service Quality Period is long enough so that well over 100 calls complete in each interval. You would also configure the tool so that only a warning would be logged if fewer than 50 calls completed in the period. For more information on configuring the Service Quality Period, Threshold, and minimum calls, see 5.3.1 Service Quality Parameters.

2.3.4 System Latencies

The Quality Advisor provides reports on various system latencies – the time taken by various system operations, or transactions – in addition to the call control latencies and user-perceived latencies described in 2.3.1 User Perceivable Failures. These system latencies do not directly contribute to the

service quality measure but often do so indirectly. For example, the time to fetch a VoiceXML page is not considered when determining service quality; however, the user-perceived response latency is considered. If an unusually latent VoiceXML page fetch occurs between user input and the subsequent prompt, then this system latency (the fetch time) will affect a user-perceived latency (the time for the system to respond to input). In fact, long user-perceived latencies are most often attributed to one or more unusually long system latencies. For this reason, the Quality Advisor tools monitors and reports on various system latencies to help track service quality failures to their source.

The Quality Advisor provides a latency reporting interface that reports on the following user-perceived and system latencies:

Call Control Latencies

- **Call Answer Latency** – The time between a call being offered to the platform and answered by the platform.

- **Call Reject Latency** – The time between a call being offered to the platform and subsequently rejected.

Prompt Latencies

- **Inbound First Prompt Latency** – The time between when an inbound call is answered by the platform to when the first prompt is played.

- **Outbound First Prompt Latency** – The time between when an outbound call is answered by the callee to when the first prompt is played.

- **Inter-Prompt Latency** – The time between the termination of prompt playback and the beginning of the following prompt playback when there is no user interaction in between. This typically measures the time between the end of playback of the last prompt on a VoiceXML page and the beginning of prompt playback on the next page. Note that `fetchaudio` is not considered as "prompt playback" for the purposes of this measurement.

- **Initial Response Latency** – For inbound calls, this is the time between the call being offered to the platform and the beginning of playback of the first prompt (essentially, the Call Answer Latency plus the Inbound First Prompt Latency). For outbound calls, this is the time between the call being answered and the beginning of playback of the first prompt (essentially, the Outbound First Prompt Latency). This provides a high-level "time to first prompt" latency that covers the underlying latencies for both inbound and outbound calls.

Response Latencies

- **Cumulative Response Latency** – The time between the completion of user interaction and the beginning of playback of the subsequent prompt. This encompasses all of the response latencies listed below.
- **DTMF Input Response Latency** – The time between the completion of DTMF input recognition and the beginning of playback of the subsequent prompt.
- **ASR Input Response Latency** – The time between the completion of ASR input recognition and the beginning of playback of the subsequent prompt.
- **No-Input Response Latency** – The time between no-input timeout and the beginning of playback of the subsequent prompt.
- **Recording Response Latency** – The time between the completion of recording and the beginning of playback of the subsequent prompt.
- **Transfer Response Latency** – The time between the completion of transfer and the beginning of playback of the subsequent prompt.

Note: For all Prompt Latencies and Response Latencies, the latency is calculated as the time between some event, and the beginning of playback of the subsequent prompt. However, if the caller or the application hangs up before a prompt begins playing, then the latency will be calculated as the time between the event and the hangup time of the call.

Fetch Latencies

- **Page Fetch Latency** – The time to fetch a VoiceXML page.
- **Audio Fetch Latency** – The time to fetch an audio prompt.
- **Grammar Fetch Latency** – The time to fetch a DTMF grammar. (ASR Grammars are fetched by the ASR engine and are not measured.)
- **Data Fetch Latency** – The time to fetch XML data using the VoiceXML `<data/>` tag.
- **JavaScript Fetch Latency** – The time to fetch a JavaScript page.

Execution Latencies

- **Page Compile Latency** – The time to compile a VoiceXML page.
- **JavaScript Execution Latency** – The time to execute a snippet of JavaScript during VoiceXML execution.

MRCP Latencies

- **MRCP ASR Session Establish Latency** – The time to establish an ASR MRCP session.
- **MRCP TTS Session Establish Latency** – The time to establish a TTS MRCP session.
- **MRCP ASR Set Params Latency** – The time to receive a response to an MRCP ASR `Set Params` request.
- **MRCP ASR Get Params Latency** – The time to receive a response to an MRCP ASR `Get Params` request.
- **MRCP TTS Set Params Latency** – The time to receive a response to an MRCP TTS `Set Params` request.
- **MRCP TTS Get Params Latency** – The time to receive a response to an MRCP TTS `Get Params` request.
- **MRCP ASR Stop Latency** – The time to receive a response to an MRCP ASR `Stop` request.
- **MRCP TTS Stop Latency** – The time to receive a response to an MRCP TTS `Stop` request.
- **MRCP Define Grammar Latency** – The time to receive a response to an MRCP ASR `Define Grammar` request.
- **MRCP Recognize Latency** – The time to receive a response to an MRCP ASR `Recognize` request.
- **MRCP Recognition Start Timers Latency** – The time to receive a response to an MRCP ASR `Recognition Start Timers` request.
- **MRCP Speak Latency** – The time to receive a response to an MRCP TTS `Speak` request.
- **MRCP Pause Latency** – The time to receive a response to an MRCP TTS `Pause` request.
- **MRCP Resume Latency** – The time to receive a response to an MRCP TTS `Resume` request.
- **MRCP Barge-In Latency** – The time to receive a response to an MRCP TTS `Barge-In` request.
- **MRCP Control Latency** – The time to receive a response to an MRCP TTS `Control` request.

2.3.5 Latency Thresholds

A threshold and a percentile can be configured for each system latency. For example, if the threshold for Page Fetch Latency is 2000 milliseconds and the percentile is 95%, then it would be expected that 95% of all page fetches complete in 2 seconds or less. The Quality Advisor will log an error if a

measured latency does not meet the configured criteria for a Service Quality Period. This error can be used to generate an SNMP trap. See 5.3.2 Latency Thresholds and 5.3.3 Alarm Notification Configuration for more details.

It is important to realize that the value for a percentile is *estimated* using the standard deviation and average for a system latency. This estimation is based on the assumption that the distribution of the latency can be modelled by a normal curve. While this methodology is usually effective, the estimate may be inaccurate if insufficient latency information was collected for a time period (for example, there were only a few calls placed to the system).

The estimation may also be flawed if the latency distribution is not well modelled by a normal distribution. This could occur if the latency distribution is *bimodal* – it has two distinct peaks. For example, the Page Fetch Latency may be bimodal if one set of pages is fetched very quickly while another set, perhaps containing pages that present the results of a lengthy database queries, normally takes a long time to return. In this case, one peak will represent the database results page fetches and the other will represent the other page fetches.

It should also be noted that the standard deviation for a system latency is usually not calculated from the raw latency data. Latency information is summarized regularly for each platform, producing a latency histogram, an average value, a minimum value, a maximum value and a standard deviation. This summary information is merged to produce summary data for the entire deployment. Unfortunately, the standard deviation cannot be calculated from summary information – the raw data is required. Instead a *pooled standard deviation* is used; this is essentially an average of the standard deviations of the sample sets. The pooled standard deviation is also used when merging latency summary information over time ranges. For example, this is done when merging hourly latency data to produce daily latency summary information. While the pooled standard deviation is often very close to the actual standard deviation, it is only an *estimation.*

In summary, the standard deviations presented for system latencies and the percentile values calculated with them are *approximations* only. While they can be used effectively in most cases, there may be scenarios where the estimated percentile values are inaccurate and a latency is erroneously flagged as not meeting its threshold criteria.

For this reason, the latency thresholds are best used as tools to narrow down service quality issues. When a system latency is not meeting its threshold criteria it will be flagged in the Quality Advisor interface and will cause errors to be logged. These errors should not be the main trigger for system alarms. Use the service quality threshold as the key metric for measuring the health of a system and to trigger alarms. Use the latency thresholds to try to understand why a deployment is not performing well when service quality issues occur.

Chapter

3 Deployment Architecture

The VoiceGenie Tools Framework consists of the following run-time components which are deployed on the VoiceGenie platform servers:

- **Log Agent** – This component is installed on the VoiceXML platform. It gathers and summarizes information logged by the platform as calls are handled.

- **Log Monitor** – This component is installed on the platform's primary Management Server. It receives periodic updates from the Log Agents containing statistics. The statistics are persistently stored in the database as they are received. The Log Monitor also receives call logs that have been gathered by the Log Agent.

- **Tools Server** – This component is a Java-based web application that is deployed along-side the System Management Console on the primary Management Server. This server implements a web services interface that serves the gathered statistic and call logs to the Call Analyst desktop client (see below). It also provides the VoiceGenie Tools Web Console, which consists of the Quality Advisor web interface as well as an administrative interface that is shared by all tools.

The Call Analyst also includes a desktop component which must be installed on the user's workstation. The Call Analyst client is a plug-in for the Eclipse IDE.

4 Installation

The VoiceGenie Tools Framework consists of run-time components which are deployed on the VoiceGenie platform servers, as well as a desktop component which is installed on a user's workstation.

4.1 Installation of Run-Time components

Like all other VoiceGenie 7.2 components, the Tools run-time components should be installed, deployed, and configured using VoiceGenie OA&M framework, which must be installed before any other VoiceGenie software can be deployed. For detailed instructions on installing these components, please refer to the *VoiceGenie 7.2 Installation Guide.*

Note: When a Log Agent is added to a cluster, it will not appear in the Service Quality Advisor or Call Analyst until Tomcat is restarted on the Tools server. The new Log Agent will still be collecting statistics for calls to the newly added platform, even before Tomcat is restarted. These statistics will only be accessible, however, after Tomcat is restarted.

4.2 Client Installation

The Call Analyst client is built on top of the open source Eclipse project (`http://www.eclipse.org`), and requires version 3.1 or above. Before un-packaging the Eclipse client, obtain the Eclipse Platform SDK from the download site: `http://www.eclipse.org/downloads/index.php`

The following packages are required to install the Call Analyst desktop client.

- Eclipse Platform SDK v3.1 or above
- `ReportingClient_v7.1_x.zip` where x is the build number.

Follow the step listed below to install and configure the client.

1. The Platform SDK is downloaded as a ZIP file. Unzip the platform SDK to a location of your choice (Note: this document assumes you unzipped the SDK to `C:\Call Analyst`).

2. Install the Call Analyst plug-in by unzipping `ReportingClient_v7.1_x.zip` into `C:\Call Analyst\eclipse`.

Note: When starting the Eclipse client for the first time after an upgrade, Eclipse should be launched with the `-clean` flag. This will refresh the plugin information in Eclipse to become up-to-date with the new version. For example, if Eclipse is installed in `C:\Call Analyst\eclipse,` it should be launched by running the command `C:\Call Analyst\eclipse\eclipse.exe -clean.`

Chapter

5 Configuration

5.1 Tools Administration Interface

The Web Interface component of the VoiceGenie Tools Framework provides a way for an administrator to manage all target applications, target permissions and subscriptions using a web browser. The interface can be accessed via the SMC at `http://server:8080/smc` where `server` is the name of the OA&M server. The default username/password on a fresh install is `pw/pw.` The Tools interface is accessed by clicking the `Tools` tab in SMC. The three management interfaces for Target Permissions, Analysis Targets, and Subscriptions can be accessed from the `Administration` menu on the left-hand side of the screen.

Note: Administrators need to log in first before they can access any of the three management interfaces.

Figure 1:   The SMC Login Screen. The Tools Web Console is accessible via the Tools tab.

5.1.1 Target Manager

The Target Manager allows an administrator to view, modify, add, and delete *analysis targets* (see 2.1.1 Analysis Target). By assigning a unique target name to applications, a user can subscribe to call logs for a specific application. This also permits the system to generate statistics specific to an application.

Viewing Targets

The Target Manager page displays a list of all defined targets in either ascending or descending alphabetical order of target name.
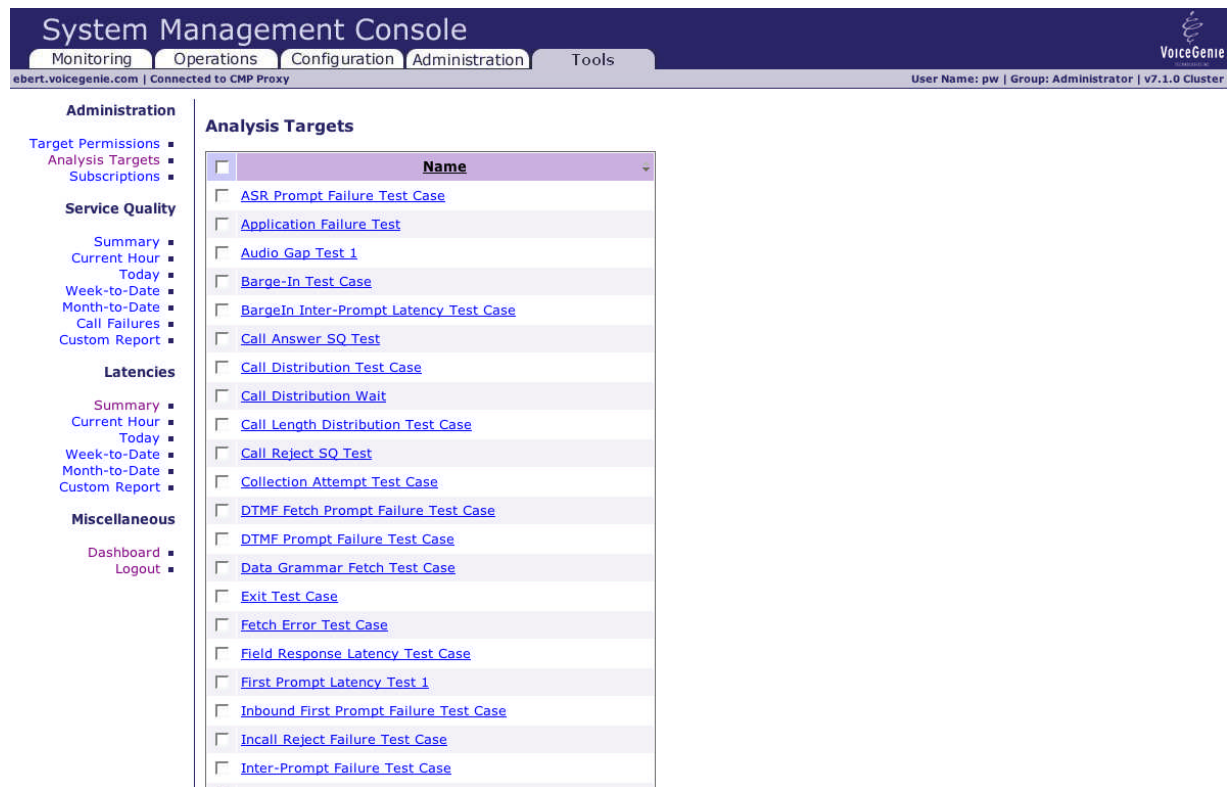
Figure 2:   The Analysis Targets Administration Screen

Deleting Targets

The administrator can delete one or more targets by clicking the check boxes for individual targets (the check box in the title line can be used to select/deselect all) and clicking on the `Delete` button. Any deleted target will also be immediately reflected on the `Target Permissions` page (the user will lose the permission of the target(s) that has been deleted), and the `Subscription Manager` page (the subscription associated with the target that has been deleted will be erased). A target can only be deleted if the platform has not collected any statistics for it.

Adding Targets

To add a new target, click on the `Create New Analysis Target` button at the bottom of the page. This will lead to a page where the new target's information can be entered.

Below is the page where a new target's information can be entered. Here is an example of a new target called `test app:`

Figure 3: Adding a new Analysis Target

Modifying Targets

To edit an existing target, go back to the `Target Manager` page and click on the name of the target to be modified. This will lead to a page where the target's information can be changed. This page looks similar to the page used to create a target and works identically except that it changes an existing target rather than creating a new one.

Note that newly added Analysis Targets, and modifications to existing Targets, only take effect after the Tools Server notifies the Log Monitor and Log Agents. This takes place at a preconfigured interval. Please see the "Location of CLC" section of the *Tools System Reference Guide,* configuration item `<clc-interval>`.

5.1.2 Target Permissions Manager

The Target Permissions Manager allows an administrator to assign Analysis Target permissions to users. Users are created in the `Administration` tab of the SMC. There are 5 types of users:

- `Administrator` – an Administrator can access data for all targets, and can manage Analysis Targets, Target Permissions and Subscriptions

- Supervisor – a Supervisor can access data for all targets, and can manage Analysis Targets
- User – a User has the same access rights as a Supervisor
- Guest – a Guest can access data for all targets
- Customer – a Customer can access the data of only the targets assigned to him/her

Customers cannot view Latency statistics because the data is not target-specific (i.e. they apply to all the targets running on a platform). Users of the other types can view Latency statistics.

When a Customer is given permissions to access one or more targets, he/she can subscribe to call logs for those targets and view statistics collected for those targets. Users of the other types can view statistics and subscribe to call logs for all targets.

Viewing Permissions

The Target Permissions page displays a list of all the users of type Customer. For each user, the user name and the list of targets accessible by the user are displayed.

Figure 4:   The Target Permissions Administration Screen

Assigning Target Permissions to Users

Click on the user name from the list of Customers to bring up the user's permissions screen. A user can be given the permission to access all the available targets by clicking the `All Targets` radio button. On the other hand, if a user is only allowed to access certain targets, click on the `Only Permitted Targets` radio button. This will show the table where target permissions can be selected. The left column lists all the targets that are currently unavailable to the user. The right column lists all of the user's permitted targets. To change permissions, click on the desired target and move them back can forth between the columns using the `Add >>` and `<< Remove` button.

Click the `Submit` button to commit the user's permission changes. It should be noted that after target permissions have been removed, subscriptions for those targets will automatically be cancelled and immediately reflected in the `Subscription Manager` page.

Figure 5:   Assigning Analysis Target permissions to a user

5.1.3 Subscription Manager

The Subscription Manager allows an administrator to view and delete subscriptions. A *subscription* allows a user to capture a specified percentage of

call logs for an Analysis Target and download them upon request. The user must be granted permissions for a target in order to subscribe to its call logs. Subscriptions are created through the Call Analyst client.

Viewing Subscriptions

The Subscription Manager page displays a list of all the subscriptions; for each one, the user name, target name, sample size, and the number of available logs are displayed. The number of available logs is the number of call logs yet to be downloaded by the subscriber. If this number exceeds the configured maximum, it is displayed with a warning icon. See the "Maintenance of Call Logs" section of the *Tools System Reference Guide* for details on the configuration item.

Figure 6:   The Subscription Administration Screen

Deleting Subscriptions

To delete subscriptions, an administrator is given the option to delete a single or multiple subscriptions simply by selecting the check boxes for individual subscriptions (the checkbox in the title line can be used to select/de-select all). Afterwards, click the Delete button and the selected subscriptions will be deleted.

5.2 Configuring the Eclipse Client

1. Launch `C:\Call Analyst\eclipse\eclipse.exe`

2. The Eclipse workbench will appear. Choose `Window > Preferences` from the main menu.
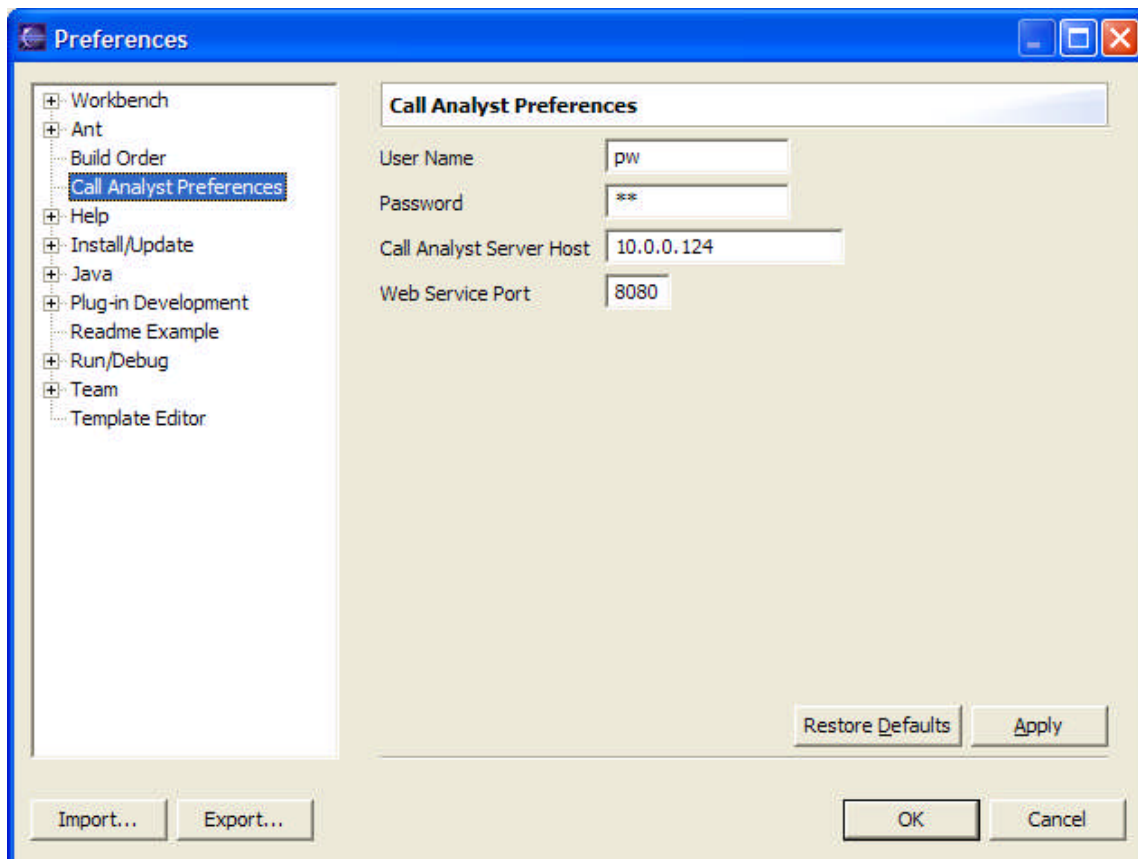
3. Select `Call Analyst Preferences` in the list on the left side of the `Preferences` page.

4. Enter your Call Analyst username and password (`pw/pw` is the default user that comes with the installation). Administrators can configure additional users by pointing their web browser to the following URL:

   `http://server:8080/smc`

   (Where `server` is replaced with the host name or IP address of the platform on which the OA&M framework is installed)

5. Specify the IP address or host name of the platform on which the Tools Server is installed.

6. Specify the port on which the Call Analyst web services are listening. The default installation uses port `8080` for all web services.

7. Press `Apply` and restart the Eclipse Platform.

5.3 Quality Advisor Configuration

All the thresholds and the service quality interval are configured through SMC. Please refer to the *OA&M Framework – SMC User's Guide* on how to change parameters in SMC. The following sections outline which component maintains the quality analyst parameters and give a brief description of each parameter.

5.3.1 Service Quality Parameters

The service quality failure thresholds are configured per Log Agent and are used to determine if there was a service quality failure for a call. The following is a list of the thresholds and a short description of them:

- `servicequality.agent.CallAnswer.threshold` – The maximum acceptable time (in ms) that the platform can take to answer a call.

- `servicequality.agent.CallReject.threshold` – The maximum acceptable time (in ms) that the platform can take to reject a call.

- `servicequality.agent.CumulativeResponse.threshold` – The maximum acceptable time (in ms) that the platform can take to play a prompt following user interaction.

- `servicequality.agent.FirstPromptInbound.threshold` – The maximum acceptable time (in ms) that the platform can take to play the first prompt on an inbound call.

- `servicequality.agent.InterPrompt.threshold` – The maximum acceptable time (in ms) that the platform can take in between consecutive prompt playbacks (with no user interaction in between).

- `servicequality.agent.FirstPromptOutbound.threshold` – The maximum acceptable time (in ms) that the platform can take to play the first prompt on an outbound call.

- `servicequality.agent.AudioGap.threshold` – The largest acceptable audio gap (in ms) that the platform can have while playing back audio to the user.

- `servicequality.agent.InboundRejectNoFailureCodes` – This is a list of strings, separated by the | character. When an inbound call is rejected by the platform with a reason code that matches one of the strings in this list, it will *not* be counted as a Service Quality failure.

- `servicequality.agent.OutboundRejectNoFailureCodes` – This is a list of strings, separated by the | character. When an outbound call is rejected by the platform with a reason code that matches one of the strings in this list, it will *not* be counted as a Service Quality failure.

The service quality threshold is configured in the Log Monitor and is used in determining when and if alarms should be raised. The following is a description of the service quality threshold:

- `servicequality.monitor.ServiceQualityThreshold` – The percentage of successful calls expected of the platform. Once the percentage of successful calls drops below this value, the Log Monitor issues an alarm. The percentage must contain 2 decimal places.

- `servicequality.monitor.MinAlertNum` – When the percentage of successful calls drops below the service quality threshold, the Log Monitor will issue an alarm. If the total number of calls is less than `MinAlertNum`, then a warning alarm will be raised, otherwise a critical alarm will be raised.

The service quality interval is configured in the Log Monitor and is transmitted to the Log Agent whenever a Log Agent or the Log Monitor starts up. The following is a description of the service quality interval:

- `servicequality.monitor.ServiceQualityPeriod` – This parameter indicates the interval at which service quality data will be forwarded to the Log Monitor so that the Log Monitor can determine the quality of the cluster. The time (in seconds), must be set to one of the following values: `300`, `360`, `600`, `720`, `900`, `1200`, `1800`, and `3600`.

5.3.2 Latency Thresholds

The latency thresholds are configured in the Log Monitor configuration but are transmitted to the Log Agents. These thresholds are used for determining whether a given latency measurement is considered acceptable. If the percentage of acceptable latency measurements for a service quality interval drops below the specified percentage, an error will be logged. The values of the latency threshold parameters are of the form `<threshold>|<alarm percentage>.` The following is a list of the threshold configuration parameters:

- `servicequality.monitor.PageFetch.LatencyThreshold` – The latency threshold configuration for the Page Fetch Latency.

- `servicequality.monitor.AudioFetch.LatencyThreshold` – The latency threshold configuration for the Audio Fetch Latency.

- `servicequality.monitor.GrammarFetch.LatencyThreshold` – The latency threshold configuration for the Grammar Fetch Latency.

- `servicequality.monitor.DataFetch.LatencyThreshold` – The latency threshold configuration for the Data Fetch Latency.

- `servicequality.monitor.JavaScriptFetch.LatencyThreshold` – The latency threshold configuration for the Javascript Fetch Latency.

- `servicequality.monitor.PageCompile.LatencyThreshold` – The latency threshold configuration for the Page Compile Latency.
- `servicequality.monitor.JavaScriptExecution.LatencyThreshold` – The latency threshold configuration for the Javascript Execution Latency.
- `servicequality.monitor.MRCPASRSessionEstablish.LatencyThreshold` – The latency threshold configuration for the MRCP ASR Session Establish Latency.
- `servicequality.monitor.MRCPTTSSessionEstablish.LatencyThreshold` – The latency threshold configuration for the MRCP TTS Session Establish Latency.
- `servicequality.monitor.MRCPASRSetParams.LatencyThreshold` – The latency threshold configuration for the MRCP ASR Set Params Latency.
- `servicequality.monitor.MRCPASRGetParams.LatencyThreshold` – The latency threshold configuration for the MRCP ASR Get Params Latency.
- `servicequality.monitor.MRCPASRStop.LatencyThreshold` – The latency threshold configuration for the MRCP ASR Stop Latency.
- `servicequality.monitor.MRCPDefineGrammar.LatencyThreshold` – The latency threshold configuration for the MRCP Define Grammar Latency.
- `servicequality.monitor.MRCPRecognize.LatencyThreshold` – The latency threshold configuration for the MRCP Recognize Latency.
- `servicequality.monitor.MRCPGetResult.LatencyThreshold` – The latency threshold configuration for the MRCP Get Result Latency.
- `servicequality.monitor.MRCPRecogStartTimers.LatencyThreshold` – The latency threshold configuration for the MRCP Recognition Start Timers Latency.
- `servicequality.monitor.MRCPSpeak.LatencyThreshold` – The latency threshold configuration for the MRCP Speak Latency.
- `servicequality.monitor.MRCPTTSSetParams.LatencyThreshold` – The latency threshold configuration for the MRCP TTS Set Params Latency.
- `servicequality.monitor.MRCPTTSGetParams.LatencyThreshold` – The latency threshold configuration for the MRCP TTS Get Params Latency.
- `servicequality.monitor.MRCPTTSStop.LatencyThreshold` – The latency threshold configuration for the MRCP Stop Latency.
- `servicequality.monitor.MRCPPause.LatencyThreshold` – The latency threshold configuration for the MRCP Pause Latency.

- `servicequality.monitor.MRCPResume.LatencyThreshold` – The latency threshold configuration for the MRCP Resume Latency.

- `servicequality.monitor.MRCPBargeIn.LatencyThreshold` – The latency threshold configuration for the MRCP Barge In Latency.

- `servicequality.monitor.MRCPControl.LatencyThreshold` – The latency threshold configuration for the MRCP Control Latency.

- `servicequality.monitor.InitialResponseTime.LatencyThreshold` – The latency threshold configuration for the Initial Response Latency.

- `servicequality.monitor.CallAnswer.LatencyThreshold` – The latency threshold configuration for the Call Answer Latency.

- `servicequality.monitor.CallReject.LatencyThreshold` – The latency threshold configuration for the Call Reject Latency.

- `servicequality.monitor.FirstPromptInbound.LatencyThreshold` – The latency threshold configuration for the Inbound First Prompt Latency.

- `servicequality.monitor.FirstPromptOutbound.LatencyThreshold` – The latency threshold configuration for the Outbound First Prompt Latency.

- `servicequality.monitor.InterPrompt.LatencyThreshold` – The latency threshold configuration for the Inter-Prompt Latency.

- `servicequality.monitor.CumulativeResponse.LatencyThreshold` – The latency threshold configuration for the Cumulative Response Latency.

- `servicequality.monitor.DTMFInputResponse.LatencyThreshold` – The latency threshold configuration for the DTMF Input Response Latency.

- `servicequality.monitor.ASRInputResponse.LatencyThreshold` – The latency threshold configuration for the ASR Input Response Latency.

- `servicequality.monitor.NoInputResponse.LatencyThreshold` – The latency threshold configuration for the No Input Input Response Latency.

- `servicequality.monitor.RecordingResponse.LatencyThreshold` – The latency threshold configuration for the Recording Response Latency.

- `servicequality.monitor.TransferResponse.LatencyThreshold` – The latency threshold configuration for the Transfer Response Latency.

5.3.3 Alarm Notification Configuration

The Log Monitor issues alarms if the service quality or latency quality drops below the configured acceptable percentages. This section describes methods of obtaining notification that an alarm has been issued.

SNMP Traps

If the system has SNMP installed on the machine running the Log Monitor, it is possible to receive an SNMP trap for each alarm that is issued. If SNMP was installed on the system before installing the Log Monitor, the VG SNMP component will be set up to send traps for each alarm. The following is a list of the alarms, a brief description of the alarm and the trap OID associated with the alarm:

- Service Quality Critical Alarm – This alarm is issued when the percentage of calls that was above the service quality thresholds (set on the Log Agents) drops below the configured acceptable service quality percentage and the number of calls is greater than or equal to the minimum alert number. The trap OID for this alarm is `.1.3.6.1.4.1.7469.251.1.314.171967468.`

- Service Quality Warning Alarm – This alarm is issued when the percentage of calls that was above the service quality thresholds (set on the Log Agents) drops below the configured acceptable service quality percentage and the number of calls is less than the minimum alert number. The trap OID for this alarm is `.1.3.6.1.4.1.7469.251.1.314.171969770.`

- Latency Error Alarm – This alarm is issued when the percentage of latencies over the whole system that are below the latency threshold drops below the acceptable pass percentage for that latency. Each latency is checked separately. The trap OID for this alarm is `.1.3.6.1.4.1.7469.251.1.314.171969186.`

6 Operations, Administration, and Maintenance

6.1 Starting and Stopping

6.1.1 Log Agent and Log Monitor

Like all other VoiceGenie 7.2 components, the System Management Console contains a page to start or stop the Log Agent and Log Monitor. Click on the `Operations` tab and click on `Start/Stop Software` on the left hand column. Click on the + icon next to the server(s) that you want to start/stop to view the components running on that server. You can click on the checkbox next to the components you want to start/stop or click the checkbox next to the server to select all components running on that server. Next, click the `Start/Stop` button to invoke the operation.

Note that the Log Agent and Log Monitor do not support the `Suspend` and `Resume` operations.

6.1.2 Tools Server

On Linux, you must be the root user to start, stop or restart the Tools Server. To become the root user log in to the system and type in `su,` then enter the root password when prompted.

Then, to start the Tools Server, issue the following command:
`/etc/init.d/vgtomcat start`

To stop the Tools Server, issue the following command:
`/etc/init.d/vgtomcat stop`

To restart the Tools Server, issue the following command:
`/etc/init.d/vgtomcat restart`

On Windows, the Tools Server can be started, stopped or restarted from the `Services` window, which can be accessed from the `Administrative Tools`

section under the Control Panel. To start the Tools Server, click on the `Apache Tomcat Service` entry and click the `Start Service` button. To stop the Tools Server, click on the `Apache Tomcat Service` entry and click the `Stop Service` button. To restart the Tools Server, click on the `Apache Tomcat Service` entry and click the `Restart Service` button.

Note that the System Management Console is also started/stopped/restarted as a side effect of starting/stopping/restarting the Tools Server.

6.2 Logging

6.2.1 Examining the Log Agent and Log Monitor Logs

The Log Agent and Log Monitor generate logging information using the VoiceGenie OA&M Framework. All logs of level `Critical` (`LOG_0`), `Error` (`LOG_1`), and `Warning` (`LOG_2`) are sent upstream and to the log file. The default log file for the Log Agent is in `/usr/local/vg-tools/logagent/log/CMP.log.logagent`. The default log file for the Log Monitor is in `/usr/local/vg-tools/logmon/log/CMP.log.logmon`.

Log levels for `Notice` (`LOG_3`) and `Information` (`LOG_4`) are stored in the log file.

`Trace` logs (`LOG_5`) are disabled by default. Enabling it will log detailed debug information. `Trace` is not recommended for a deployment environment because it floods the trace files quickly and decreases system performance.

To enable trace, go to the Log Agent or Log Monitor Configuration in the System Management Console and set `cmp.trace_flag` to `true`. Click `Update` to submit the configuration change.

6.2.2 Examining Tools Server Logs

By default, there are three log files generated by the Tools Server:

1. `webInterface.log:` information logged from the Tools Server when Analysis Targets, subscriptions, and/or users are modified.

2. `reportService.log:` debugging information logged while generating statistical reports.

3. `webserviceSOAP.log:` debugging logs of the SOAP messages communicated to and from the Web Service request. By default, SOAP debugging logs are disabled, i.e. `log4j.category.com.voicegenie.tools.webservices.impl.SOAPLoggingHandler` is set to `OFF` in `log4j.configuration`. Debug logs can be enabled by replacing `OFF` with `DEBUG`.

By default, all of the log files will be located at
`$CA_INSTALL_ROOT/server/logs`. However, the location can be changed
in the `Log4J` configuration file found at `$CA_INSTALL_ROOT/server/WEB-INF/classes/log4j.properties.` Please see the *System Reference Guide*
for details.

6.3 Cleaning the Log Monitor error directories

The Log Monitor directory structure contains two subdirectories for saving
data when the system is not functioning normally. The subdirectories are
located in `/usr/local/vg-tools/logmon/logs` (under Linux) or
`C:\VoiceGenie\vg-tools\logmon\logs` (under Windows). The `tx-errors` directory contains failed database transactions, serialized in XML
format. The `calllogerrors` directory contains call logs from subscriptions
that could not be transformed into the Call Analyst XML format.

These directories should remain empty during normal operation of the Log
Monitor, however, it is important to clean these directories if they become full.
Due to filesystem limitations, performance and stability may be affected if the
number of files exceeds 64000. System administrators should occasionally
clean these directories, especially if there have recently been issues with the
database server.

![Genesys - An Alcatel-Lucent Company logo]

Chapter

7 Using the Call Analyst Client

7.1 Launching the Call Analyst Perspective

1. Open Eclipse (e.g. `C:\Call Analyst\eclipse\eclipse.exe`).

2. Open the Call Analyst Perspective by selecting `Window > Open Perspective > Other` from the main menu. Select `Call Analyst Perspective` from the dialog that appears and click `OK.`
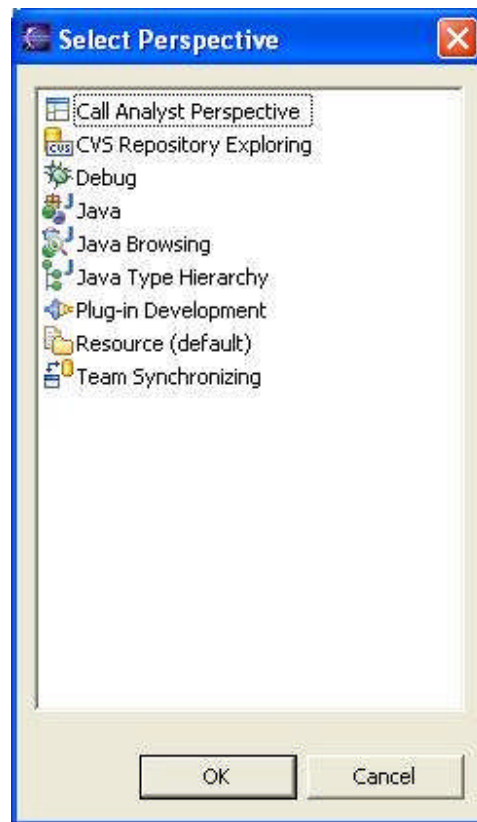
Figure 7:   Selecting the Call Analyst Perspective

7.2 Getting Started

1. Once you have installed the Call Analyst (see above), you should first familiarize yourself with the basic concepts upon which the tool is based by reading 1.2.2 Quality Advisor of this document.

2. Make sure you have a user name assigned (an administrator must do this for you in SMC) and that you have properly configured the Call Analyst Client (see 4.2 Client Installation).

3. Make sure your applications have been assigned to analysis targets. This must be done by an administrator through the Call Analyst Admin Interface (see Adding Targets). Your applications must either contain the appropriate application meta tag or be assigned to an appropriate DNIS value for the mapping to take effect.

4. Make sure that the contexts in your application can be uniquely identified. This should require no action or modifications for most applications.

5. Run Call Analyst and launch the Call Analyst Perspective (see 2.2 Call Analyst).

6. Read the detailed instructions on using the Call Analyst client to view statistics and call logs. These instructions can be found by accessing the Call Analyst Help files using the Help Reader that is launched from the `Help > Contents` menu item.

7. Create a new workbook using the New Workbook Wizard. The wizard can be launched from `File > New > Project...`

8. Create any required subscriptions for your analysis targets using the Call Log Subscription View. *100% subscriptions are generally not desirable for live deployments under load* as they will unnecessarily stress the system and produce far more data than could ever be analyzed.

9. After, the application has been run for a period of time, generate analysis reports using the Report Generation View to get insight into how your application is executing.

10. Update your subscriptions using the Call Log Subscription View to get call logs and view them in the Call Log Editor to investigate usability issues that appear in the analyses.

Chapter

8 Using the Quality Advisor Web Interface

The Quality Advisor web interface is an integrated part of the VoiceGenie Tools Web Console. This section will discuss the features of the web console that pertain to the Quality Advisor interface. For information on logging into the Tools Web Console and using the administration features, see 5.1 Tools Administration Interface.

8.1 General layout

The Quality Advisor web interface provides a set of reports on the Service Quality and System Latency metrics that are calculated by the tool. All pages in the interface share a common layout that we will describe in this section.

8.1.1 Menu Sidebar

On the left hand side of the screen, a menu sidebar provides easy navigation between parts of the Interface. This menu is divided into the following sections:

- `Administration` – This section allows Analysis Targets, Target Permissions and call analyst log subscriptions to be managed. This menu is described in 5.1 Tools Administration Interface.

- `Service Quality` – This section provides reports on Service Quality metrics that can be filtered based on analysis target, machine and time. This section is available to all user levels; however users may only view data for analysis targets for which they have permissions. This menu contains the following links:

  - `Summary` – This links to the `Service Quality Summary` screen described in 8.3 The Service Quality Summary Screen.

  - `Current Hour` – This links to `Service Quality Details` screen described in 8.4 The Service Quality Details Screen. This page

displays Service Quality information for the current hour, for all platforms and for all analysis targets (for which the user has permissions).

- `Today` – This links to `Service Quality Details` screen described in 8.4 The Service Quality Details Screen. This page displays Service Quality information for the current day, for all platforms and for all analysis targets (for which the user has permissions).

- `Week-to-Date` – This links to `Service Quality Details` screen described in 8.4 The Service Quality Details Screen. This page displays Service Quality information for the current week, for all platforms and for all analysis targets (for which the user has permissions).

- `Month-to-Date` – This links to `Service Quality Details` screen described in 8.4 The Service Quality Details Screen. This page displays Service Quality information for the current month, for all platforms and for all analysis targets (for which the user has permissions).

- `Call Failures` – This links to the `Call Failures Listing Request` screen described in 8.6 The Call Failures Listing Request Screen.

- `Custom Report` – This links to the `Custom Service Quality Report Request` screen described in 8.9 The Custom Service Quality Report Request Screen.

- `Latencies` – This section provides reports on latency measurements that can be filtered based on machine and time. This section is only available to Administrators, Supervisors, Users and Guests. The menu contains the following links:

  - `Summary` – This links to the `Latency Summary` screen described in 8.11 The Latency Summary Screen.

  - `Current Hour` – This links to the Latency Details screen described in 8.12 The Latency Details Screen. This page displays system latency information for the current hour, and for all platforms.

  - `Today` – This links to the `Latency Details` screen described in 8.12 The Latency Details Screen. This page displays system latency information for the current day, and for all platforms.

  - `Week-to-Date` – This links to the `Latency Details` screen described in 8.12 The Latency Details Screen. This page displays system latency information for the current week, and for all platforms.

  - `Month-to-Date` – This links to the `Latency Details` screen described in 8.12 The Latency Details Screen. This page displays system latency information for the current month, and for all platforms.

- `Custom Report` – This links to the `Custom Latency Report Request` screen described in 8.14 The Custom Latency Report Request Screen.
- `Miscellaneous` – This section provides some miscellaneous features such as changing passwords and logging out. This section is available to all user levels. The following links are available in this menu:
    - `Dashboard` – This link displays the Tools Dashboard.
    - `Logout` – This link logs the user out of the system.

8.1.2 Report Preamble

The top portion of the report screens consists of a header that is devoted to displaying the filter parameters used to construct the displayed report. The following headings may appear in the preamble:

- `Range` – This is the time range that was used to query the presented data.
- `Granularity` – This is the time granularity of the presented data. It can be hours, days, weeks or months.
- `Analysis Target` – This is a list of the analysis targets to which the data applies. Unless the report breaks the data down by analysis target, the presented data is an aggregate of the data for all of the listed analysis targets. Note that the listed analysis targets are hyperlinks; clicking on an analysis target will show the data relevant to the selected analysis target only.
- `Platforms` – This is a list of the platforms to which the data applies. Unless the report breaks the data down by platform, the presented data is an aggregate of the data for all of the listed platforms. Note that the listed platforms are hyperlinks; clicking on a platform will show the data relevant to the selected platform only.
- `Failure Types` – Reports that list failures that occurred in a specific time period can be filtered by failure type (see 2.3.2 Service Quality). This lists the failure types that were used to filter the data displayed. Note that the listed failure types are hyperlinks; clicking on a failure type will show the data relevant to the selected failure type only.
- `Group by` – Some reports allow the data to be displayed and grouped by either platform or analysis target. This header shows which of the two was selected when requesting the data.

The `Analysis Target` and `Platform` lists may be truncated if they are more than a couple lines long. In this case an ellipsis (…) will appear at the end of the list. By clicking on the + icon at the front of the list, it can be expanded to its full length. The icon will change to a - icon after it is clicked; it can then be used to collapse the list to its original length.

Custom Service Quality Report

Range	[2006-05-01 00:00:00 - 2006-06-01 00:00:00]	**Granularity** [MONTHS] **Group By** [Analysis Target]

Analysis Targets ⊟ Barge-In Test Case   Call Length Distribution Test Case   Call Reject SQ Test
Collection Attempt Test Case   DTMF Fetch Prompt Failure Test Case   First Prompt Latency Test 1
Hello World   Long Init Test Case

Platforms   10.0.0.54

Figure 8:   Example of Report Preamble

8.1.3 A note about bookmarking reports and graph pages

The Tools Web Console was designed to allow reports to be saved with web browser bookmarks. That is, all the information required to generate a report is sent in the URL of each page request. However, due to the complexity of the Service Quality Graph and Latency Graph, these pages cannot be bookmarked. The graph pages require data that is stored in the session, therefore it is not possible to generate a graph by making a request to the URL alone. The graphs must be generated through the other reports available through the Tools Web Console.

8.2 The Tools Dashboard

Upon logging into the SMC and clicking on the `Tools` tab, the user is presented with the Tools Dashboard. The Dashboard can also be accessed from the `Dashboard` link under `Miscellaneous` on the menu at the left-hand side of the screen.

The Dashboard is a summary of the various Call Analyst and Service Quality Advisor analyses. The statistics displayed are aggregated from the media platforms on the deployment, for the Analysis Targets that the user has permissions to. The report preamble displays the list of targets that the statistics are collected from.

The Dashboard reports on three categories of data for the current hour, day, week and month:

- `Service Quality` – this section summarizes the overall performance of the deployment:
  - `# Calls` – the total number of calls handled by the system during the time period
  - `Avg Len` – the average length per call
  - `SQ%` – the Service Quality for the time period. If this is below the configured Service Quality threshold (see 5.3.1 Service Quality Parameters), it is highlighted in red and a warning icon is displayed beside it

- **Recognitions** – this section summarizes the performance of ASR and DTMF recognition:
    - **#** – the total number of recognitions executed during the time period
    - **Avg Tries** – the average number of recognition attempts per successful recognition
    - **Voice [Match]** – the total number of ASR recognition attempts, along with the percentage of these that recognized successfully. For example **133 [80.5%]** means that there were 133 ASR attempts, 80.5% of which were successful.
    - **DTMF [Match]** – same as the **Voice [Match]** column, for DTMF recognitions
- **Transfers** – this section summarizes the transfers that occurred over the time period
    - **#** – the total number of transfer attempts
    - **Conn'd** – the number of transfers that were successfully connected
    - **Avg Dur** – the average duration of the successful transfers

Figure 9:   Tools Dashboard

8.3 The Service Quality Summary Screen

The `Service Quality Summary` screen can be accessed at any time from the `Summary` link under `Service Quality` on the menu at the left-hand side of the screen.

This page provides a summary of the current Service Quality metrics for all machines in the deployment and all analysis targets for which the user has permissions. By clicking one of the analysis targets or platforms in the report preamble, the same information can be viewed for selected target or platform only.

Figure 10: The Service Quality Summary Screen

Data is shown for the current hour, current day, current week and current month. For each of these time periods, the following information is provided in a table:

- `Time Range` – This identifies the time period to which the row applies. This is either: `This Hour`, `Today`, `This Week`, or `This Month.`

- `Calls` – This is the number of calls that *ended* in the specified time period.

- `Failed Calls` – This is the number (and percentage) of calls that ended during the specified time period and experienced a user perceivable failure.

- `SQ` – This is the Service Quality for the specified time period. If this is below the configured Service Quality threshold, it is highlighted in red and a warning icon is displayed beside it.
- `Call Control Failures` – This presents the number of Call Control Failures that occurred in the specified time period as well as the percentage of failures that were of this type.
- `UP-Latency Failures` – This presents the number of User-Perceived Latency Failures that occurred in the specified time period as well as the percentage of all failures that were of this type.
- `App Errors` – This presents the number of Application Errors that occurred in the specified time period as well as the percentage of failures that were of this type.
- `System Errors` – This presents the number of System Errors that occurred in the specified time period as well as the percentage of failures that were of this type.
- `Details` – This provides a link to the `Service Quality Details` page for the specified time period.

Note that since a single call may experience multiple failures, the total number of failures may be more than the number of failed calls. Also note that the failure information is presented as a hyperlink; clicking on the link shows detailed information about the individual failures using the Call Failures report (see 8.6 The Call Failures Listing Request Screen).

Note that at the beginning of a month, the number of calls included in the monthly report may be less than the number of calls in the weekly report if the month began in the middle of the week. This sometimes throws off users as they might expect the monthly figures to always include more calls than the weekly figures. This is expected behavior however.

8.4 The Service Quality Details Screen

This screen provides detailed information on the Service Quality for a specific period of time. By clicking on the `Current Hour`, `Today`, `Week-to-Date`, or `Month-to-Date` links in the `Service Quality` menu, the `Service Quality Details` screen will be displayed for the selected period of time. By default, the screen shows aggregated information for all analysis targets for which the user has permissions. It also shows information for all platforms; however the aggregated value is shown as well as the breakdown by platform. By clicking one of the analysis targets or platforms in the report preamble, the same information can be viewed for selected target or platform only.
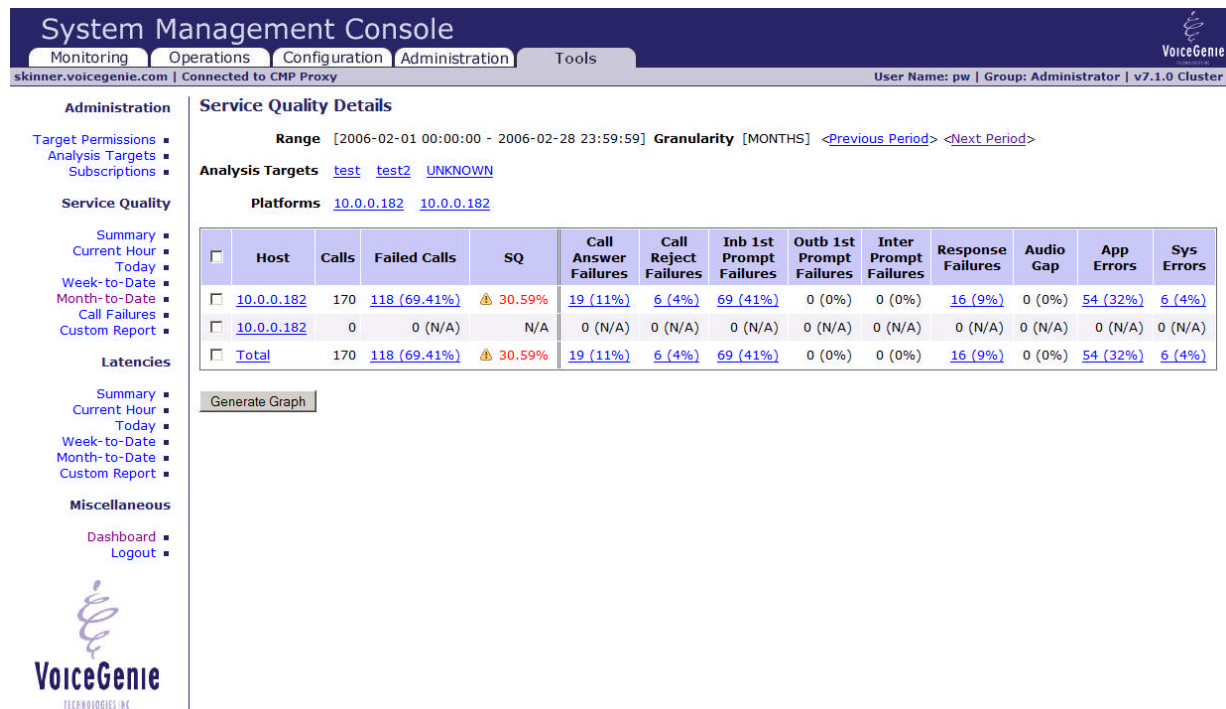
Figure 11: The Service Quality Details Screen

The screen contains a table that displays the following information for the time period indicated in the preamble:

- `Host` – The platform to which the information applies. If the displayed host name is `Total` then this information is aggregated for all platforms that appear in the preamble.

- `Calls` – This is the number of calls that *ended* in the specified time period.

- `Failed Calls` – This is the number (and percentage) of calls that ended during the specified time period and experienced a user perceivable failure.

- `SQ` – This is the Service Quality for the specified time period. If this is below the configured Service Quality threshold, it is highlighted in red and a warning icon is displayed beside it.

- `Call Answer Failures` – This is the number of Call Answer Failures experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `Call Reject Failures` – This is the number of Call Reject Failures experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `Inb 1st Prompt Failures` – This is the number of Inbound First Prompt Failures experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `Outb 1st Prompt Failures` – This is the number of Outbound First Prompt Failures experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `Inter Prompt Failures` – This is the number of Inter-Prompt Failures experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `Response Failures` – This is the number of Cumulative Response Failures experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `Audio Gap` – This is the number of Audio Gap errors that were experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `App Errors` – This is the number of Application Errors that were experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

- `Sys Errors` – This is the number of System Errors that were experienced by calls that ended in the specified time period as well as the percentage of failures that were of this type.

Note that since a single call may experience multiple failures, the total number of failures may be more than the number of failed calls. Also note that the failure information is presented as a hyperlink; clicking on the link shows detailed information about the individual failures using the Call Failures report (see 8.6 The Call Failures Listing Request Screen).

Each row in the table contains a check box in the left-most column. This check box allows a row of data to be selected for graphing. The check box in the top-left corner can be used to select or unselect all rows in a single click. Once the desired rows have been selected, a graphical representation of failed and successful calls and the service quality is displayed (see 8.8 The Service Quality Graph Screen for more details). When the graph is generated, the data is displayed at a lower time granularity than is used in this screen. This is done to show how the data is broken down within the current time period. The granularity of the graph is based on the granularity used in the `Service Quality Details Screen` as follows:

- `Month` – The graph displays all days within the month (for which there is data).

- `Week` – The graph displays all days within the week (for which there is data).

- `Day` – The graph displays all hours within the day (for which there is data).

- `Hour` – The graph simply displays the data for the selected hour. Arguably this is not a very useful graph; however, it is included for completeness.

Graphing data generated with a custom report can produce graphs that span an arbitrary period of time and granularity (e.g. two weeks of daily data) if this is required.

Clicking on the value in the `Host` column shows how the data is broken down within the current time period. This is similar to the graphs described above except that the data is displayed in tabular form. See 8.5 The Service Quality Breakdown Screen for more details.

At the top right hand side of the preamble section of the screen there are two links: `<Previous Period>` and `<Next Period>`. Clicking these links will show the identical report for the previous or next period of data at the current granularity. For example, if you are looking at today's Service Quality information and click on `<Previous Period>,` you will be shown yesterday's Service Quality information. Only the time period of the report will be changed, the time granularity, analysis targets, and platforms will remain the same.

8.5 The Service Quality Breakdown Screen

This screen is accessed from the `Service Quality Details` screen (see 8.4 The Service Quality Details Screen) by clicking on the `Host` value for a row in the results table (this value is either a platform name, or `Total` which indicates that the row contains aggregate data for all platforms in the table). This screen shows a breakdown of the service quality information for that row over time. The granularity of the `Service Quality Breakdown` screen is based on the granularity used in the `Service Quality Details` Screen as follows:

- `Month` – The breakdown displays all days within the month (for which there is data).
- `Week` – The breakdown displays all days within the week (for which there is data).
- `Day` – The breakdown displays all hours within the day (for which there is data).
- `Hour` – The breakdown simply displays the data for the selected hour. Arguably this is not a very useful breakdown; however, it is included for completeness.

The breakdown is displayed in both a tabular form and graphical form. The table contains most of the same columns as the `Service Quality Details` screen except that each row contains data for a time period instead of a host. The time period associated with the Service Quality data is indicated in the `Time` column.

The graph below the table shows the number of successful and failed calls as well as the Service Quality for each time period in the table. The format of the graph is identical to that used in the `Service Quality Graph` screen described in 8.8 The Service Quality Graph Screen.
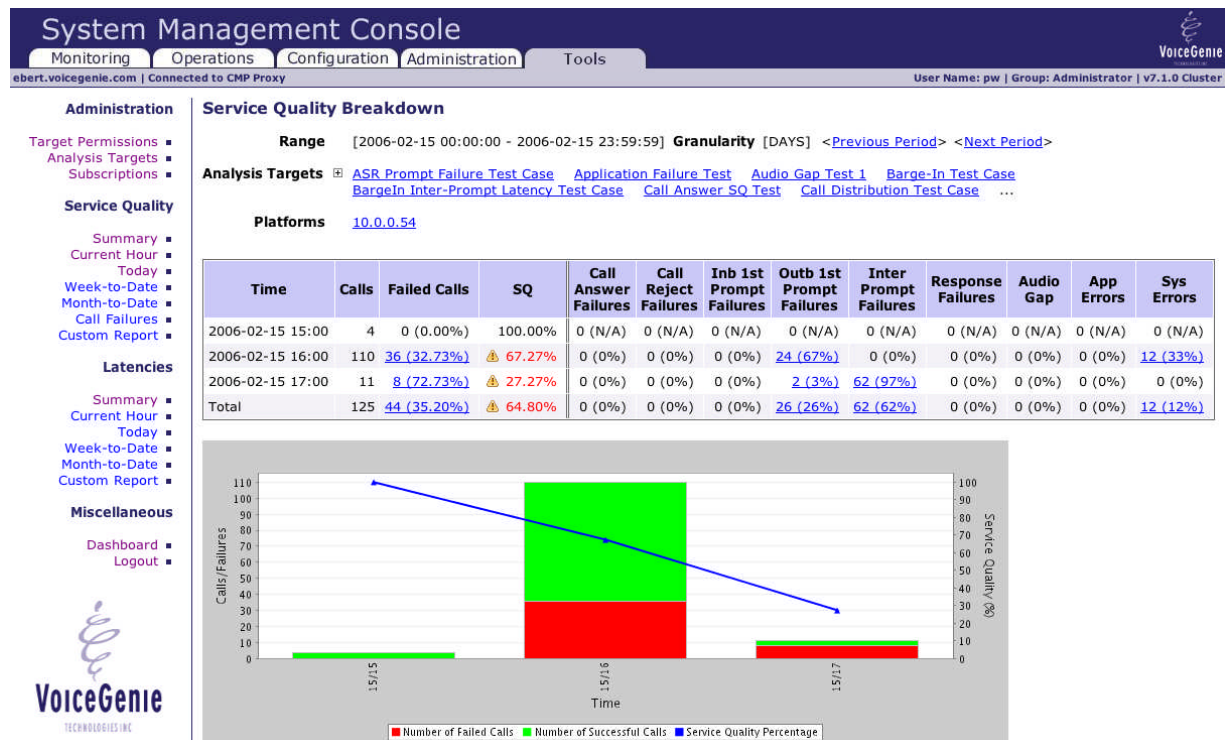
Figure 12: The Service Quality Breakdown Screen

8.6 The Call Failures Listing Request Screen

This screen is used to create a custom report providing detailed information about the failures that were detected within a specified time period. The failure information is displayed using the `Call Failures` Screen (see 8.7 The Call Failures Listing Screen).

Figure 13: The Call Failures Listing Request Screen

The `Call Failures Listing Request` Screen provides a number of fields and options that are used to parameterize a query for a set of call failure information. The parameters that can be set are as follows:

- `From`, `To` – Call failures are reported to the Quality Advisor at the end of each Service Quality Period for all calls that ended within that period. These parameters set the lower and upper bounds on the time interval used to filter the reported call failures. Failures that were reported in a Service Quality interval that ended within the time interval specified by the `From` and `To` parameters will be included in the query results.

- `Failure Types` – This parameter can be used to restrict the query results to one or more failure types. Select `All Failure Types` to include all types in the results. If `Select failure types` is clicked, then a list of failure types will appear. One or more failure types can be selected from the list (Hold down the `Ctrl` key to select multiple types).

- `Analysis Targets` – This parameter can be used to restrict the query results to calls that have been assigned to specific Analysis Targets. Select `All permitted analysis targets` to see results for all analysis targets for which the current user has permissions. If `Select analysis targets` is clicked, then a list of analysis targets will appear. One or more analysis targets can be selected from the list (Hold down the `Ctrl` key to select multiple targets).

- `Platforms` – This parameter can be used to restrict the query results to calls executed on specific platforms. Select `All platforms` to see results for all platforms. If `Select platforms` is clicked, then a list of platforms will appear. One or more platforms can be selected from the list (Hold down the `Ctrl` key to select multiple platforms).

- `Items per page` – This parameter does not affect the contents of the query results but instead changes the way they are presented. This parameter specifies how many failures should be presented on each page of the `Call Failures` Screen. The user is able to navigate trough the various screens containing the query results using `next` and `previous` buttons on the `Call Failures` Screen (see 8.7 The Call Failures Listing Screen for more details).

When all options have been selected, click the `Submit` button to submit your query.

8.7 The Call Failures Listing Screen

This screen is used to display information about call failures that were detected in a specific time period. It is important to note that call failures are reported to the Quality Advisor at the end of each Service Quality Period. Each report contains failure information for all calls that ended within that period. Call Failures can only be queried by the time at which they were *reported* not by the time at which the failure occurred. This can have some unexpected implications. For example, consider a call that experiences a failure in one hour, say at 3:55pm, but ends in the following hour, say at 4:05pm. The failure that occurred at 3:55pm will not be listed if a query requests call failure listings from 3:00pm to 4:00pm. Instead, it will show up when querying for call failure listings from 4:00pm to 5:00pm.

While this may seem counter-intuitive, it makes it easier to request the failures that affected the Service Quality metric for a specific period of time. To prevent a call that experiences multiple failures from being counted twice, a failed call only affects the service quality period in which it ends. Thus, by querying for call failure listings in this way, we are able to find all failures which affected the service quality metrics for the time period in question.

Since calls rarely, if ever, last longer than an hour, the time range of the query can be expanded in order to include calls that ended in later periods. For example, to find all failures that occurred between 3:00pm and 4:00pm, the call failure listings query should request failure listings from 3:00pm to 5:00pm or even 6:00pm. Call failures which occurred outside of the desired time range can easily be ignored by the user by sorting the data by failure time.

The `Call Failure Listing` Screen will display call failure listings for the time range, analysis targets, platforms and failure types stated in the report preamble. Clicking on the analysis target, platform, or failure type will narrow down the search to the analysis target, platform or failure type selected.

Figure 14: The Call Failures Listing Screen

The `Call Failure Listing` table displays individual call failures. Note that calls can have more than one Service Quality failure. This explains why the number of items in the `Call Failure Listing` can be greater than the number of failed calls indicated on the `Service Quality Summary` or `Details` screens. The following information is displayed for each failure:

- `Detection Period` – This is the start time of the Service Quality period in which the failed call ended.

- `Failure Time` – This is the time at which the failure occurred.

- `Call Id` – This is the VoiceGenie session id for the call.

- `Analysis Target` – This is the analysis target that the call was assigned to.

- `Platform` – This is the platform on which the call executed.

- `Duration` – This is the duration of the failed call.

- `Failure Type and Info` – This is the type of failure that occurred, followed by a brief description of the failure. The description has a different meaning for each failure type:
  - Call Control Latency Failures and User-Perceived Latency Failures (see Call Control Latency Failures and User-Perceived Latency Failures): The latency time that the caller experienced (in

milliseconds) exceeded the configured threshold, and is displayed here.
E.g. `Call Answer Failure : 10789 ms`

- Application Errors (see Application Error): The text content of the
  `<log>` tag with the failure label is displayed. E.g. `Application
  Error : A database error occurred.` The string is truncated to
  128 characters if it is too long.

- System errors (see System Error): An error message is displayed,
  depending on the error:

  - `Inbound call error` or `Outbound call error` if a call ends
    with a `syserr` reason code

  - `Inbound reject (code)` if an inbound call is rejected with
    reason code

  - `Outbound reject (code)` if an outbound call is rejected with
    reason code

  - `Abnormal termination` if the call is forced to terminate

  - `Call dropped` if the call does not reach a completed state

Only a limited number of failures are displayed on each page; the number of
failures is configured using the `Items per page` parameter when making a
query. The controls above the failure listing can be used to navigate between
data pages. The controls work as follows:

- `First` – This jumps to the first page of data.

- `Prev` – This jumps to the previous page of data.

- `1,2,3,...` – Clicking on the nth number jumps to the nth page of data.

- `Next` – This jumps to the next page of data.

- `Last` – This jumps to the last page of data. (Note that the last page can
  only be accessed via this link, i.e. there is no numbered link for the last
  page.)

The failure listing can be sorted by any of the columns in the listing table. To
sort the list using the data in a column, click on the column heading. The
header of the column that is being used to sort the data will be a darker shade
than the other headers. The arrow on the right hand side of the header indicates
the order in which the data is sorted. If the arrow points upwards, the data is
sorted in increasing order; if the arrow points downwards, the data is sorted in
decreasing order.

Note that if the query produces a list of over 1,000 failures, then only the first
1,000 items will be displayed. The list of failures can be viewed in smaller
increments by refining the search criteria using the Call Failure Listing
Request.

8.8 The Service Quality Graph Screen

This screen provides a visualization of the number of failed and successful calls as well as the Service Quality over a period of time. This screen can be reached by selecting data for graphing from either the `Service Quality Details` screen or the `Custom Service Quality Report` Screen. The graphs on this screen may be organized by platform or analysis target, depending on how the data was organized in the original report.

Each graph shows both the number of successful and failed calls as well as the Service Quality for one or more time periods. The time periods are listed along the horizontal axis. For each time period, a vertical bar represents the number of calls processed. The green portion of the vertical bar represents the calls that were successful; the red portion represents the calls that failed. The triangular points at each time period connected by a solid line represent the Service Quality level for the period. The left hand side of the graph labels the vertical axis by numbers of calls; the right hand side labels the vertical axis by Service Quality percentage.
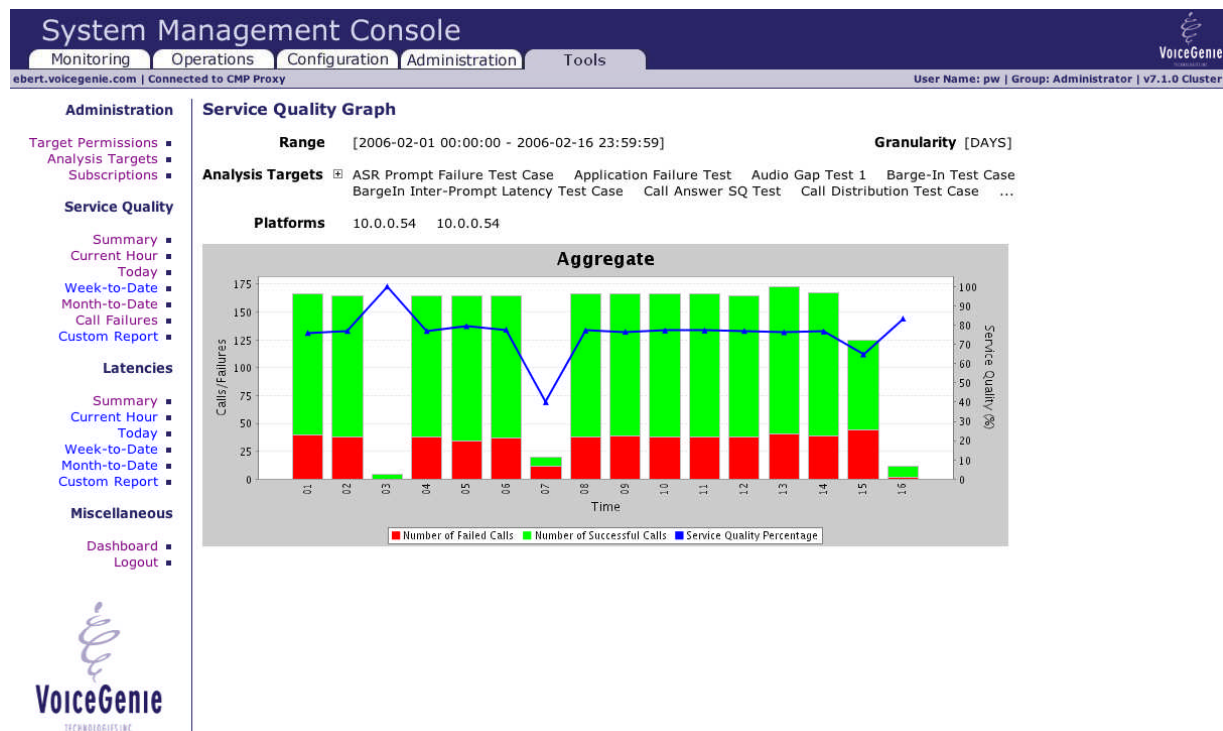
Figure 15: The Service Quality Graph Screen

Moving the mouse cursor over a vertical bar or one of the service quality points displays a tooltip that provides the total number of calls, the number of failed calls as well as the service quality for the applicable time period.

Clicking on the vertical bar for a time period shows the Call Failure Listing for the specified time period. The Call Failure Listing is filtered using the same parameters (analysis target and platform) used to generate the original graph.

8.9 The Custom Service Quality Report Request Screen

This screen allows the user to request a custom service quality report. The user is presented with a set of fields and options that can be used to parameterize a query for detailed service quality information. The available parameters are as follows:

- `From, To` – These parameters set the bounds for the time range for which service quality information will be requested. Times must be entered in the following format: `YYYY-MM-DD hh:mm:ss` (for example, `2005-08-12 12:00:00`). The button to the right of the text fields can be used to select a time using a calendar widget.

- `Granularity` – This defines the time granularity for the service quality data. Reports can be requested on an hourly, daily, weekly or monthly basis.

- `Group by` – The presented service quality data can be presented and grouped by either Analysis Target or by Platform. If `Analysis Target` is selected then the information will be broken down by Analysis Target only. Data will not be presented for individual platforms; instead, aggregated data will be included for those platforms specified in the Platforms parameter (see below). Conversely, if `Platform` is selected then the information will be broken down by platform only and data will not be presented for individual Analysis Targets.

- `Analysis Targets` – This parameter can be used to restrict the query results to calls that have been assigned to specific Analysis Targets. Select `All permitted analysis targets` to see results for all analysis targets for which the current user has permissions. If `Select analysis targets` is clicked, then a list of analysis targets will appear. One or more analysis targets can be selected from the list. (Hold down the `Ctrl` key to select multiple targets.)

- `Platforms` – This parameter can be used to restrict the query results to calls executed on specific platforms. Select `All platforms` to see results for all platforms. If `Select platforms` is clicked, then a list of platforms will appear. One or more platforms can be selected from the list (Hold down the `Ctrl` key to select multiple platforms).

When you are satisfied with your query parameters, click on the `Submit` button to perform your query.

Figure 16: The Custom Service Quality Report Request Screen

8.10 The Custom Service Quality Report Screen

This screen presents the results of Service Quality Custom Report queries made using the `Customer Service Quality Report Request` screen described in 8.9 The Custom Service Quality Report Request Screen. This screen provides the same information as the `Service Quality Details` screen; however the data can be broken down by either Analysis Target or Platform. The way in which the information is broken down and ordered is controlled by the value of the `Group By` parameter used in the request.

If the data is ordered by Analysis Target, then the data is shown in multiple tables, one for each target specified in the query. The heading above the table indicates the Analysis Target to which the data applies. The table at the bottom of the page labeled `Aggregate` shows aggregated data for all analysis targets. Each table includes data for all platforms indicated in the report preamble; the data is not broken down by platform.

If the data is ordered by Platform, then each table shows information for a specific platform. The table at the bottom of the page labeled `Aggregate` show aggregated data for all platforms listed in the report preamble. Each table

includes data for all Analysis Targets indicated in the report preamble; the data is not broken down by Analysis Target.

Figure 17: The Custom Service Quality Report Screen (Grouped By Analysis Target)

The meaning and significance of the information presented is identical to the data table used in the `Service Quality Details` screen. See 8.4 The Service Quality Details Screen for more details.

8.11 The Latency Summary Screen

This page provides a summary of the current Latency metrics for all machines in the deployment. Latency information cannot be broken down by analysis target. By clicking on one of the platforms in the report preamble, the same information can be viewed for the selected platform only. This screen can be accessed at any time from the `Summary` link under `Latencies` on the menu at the left-hand side of the screen.
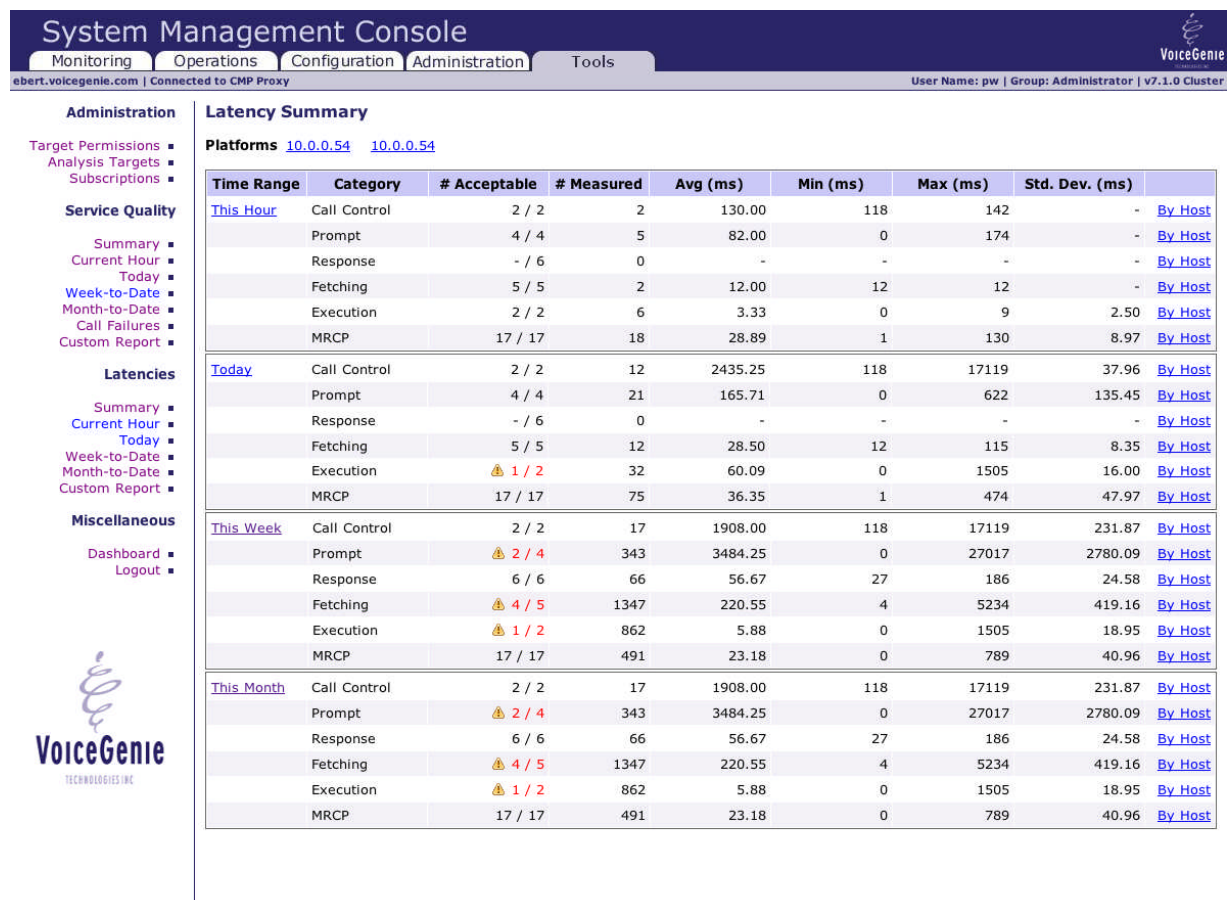
Figure 18: The Latency Summary Screen

Data is shown for the current hour, current day, current week and current month in four subsections of the table. The latency information is summarized by category (see 2.3.4 System Latencies for a description of the latencies in each category). For each of these time periods, the following information is provided:

- `Time Range` – This is the time range that the data in this section of the table applies to. This is either: `This Hour`, `Today`, `This Week`, or `This Month`.

- `Category` – This lists the various categories of latencies for which we have data. This list is repeated for each time range. The data in each row applies to the time range and category combination.

- `# Acceptable` – This indicates the number of latencies in the category that are passing their threshold criteria (see 2.3.5 Latency Thresholds). Two numbers are presented, separated by a slash. The first number is the number of latencies that pass the threshold criteria; the second number is the total number of latencies in the category. If any latencies in the

category are not passing the threshold criteria, the data is shown in red and a warning icon is displayed beside it.

- `# Measured` – This is the number of latency measurements that were taken for all latency types in the category.

- `Avg (ms)` – This is the average latency value for the category in milliseconds. This average is calculated using all latency types in the category.

- `Min (ms)` – This is the minimum latency value for the category in milliseconds. This minimum is calculated using all latency types in the category.

- `Max (ms)` – This is the maximum latency value for the category in milliseconds. This maximum is calculated using all latency types in the category.

- `Std. Dev. (ms)` – This is the pooled standard deviation value for the category. This value is calculated with a pooled standard deviation function using data from all latencies in the category.

The `By Host` link at the end of each row in the table shows detailed information on the latencies in the category broken down by host. See 8.13 The Latency Details By Host Screen for more details.

Clicking on the time range values (`This Hour`, `Today`, `This Week` or `This Month`) displays detailed statistics for all of the individual latencies for the time range selected. See 8.12 The Latency Details Screen for more information on the `Latency Details` Screen.

8.12 The Latency Details Screen

This screen presents statistics on the individual latencies for the time range and platforms specified in the report preamble. The information is presented in a table that is broken into subsections – one for each latency category. Each row of the table provides information about a single system latency.
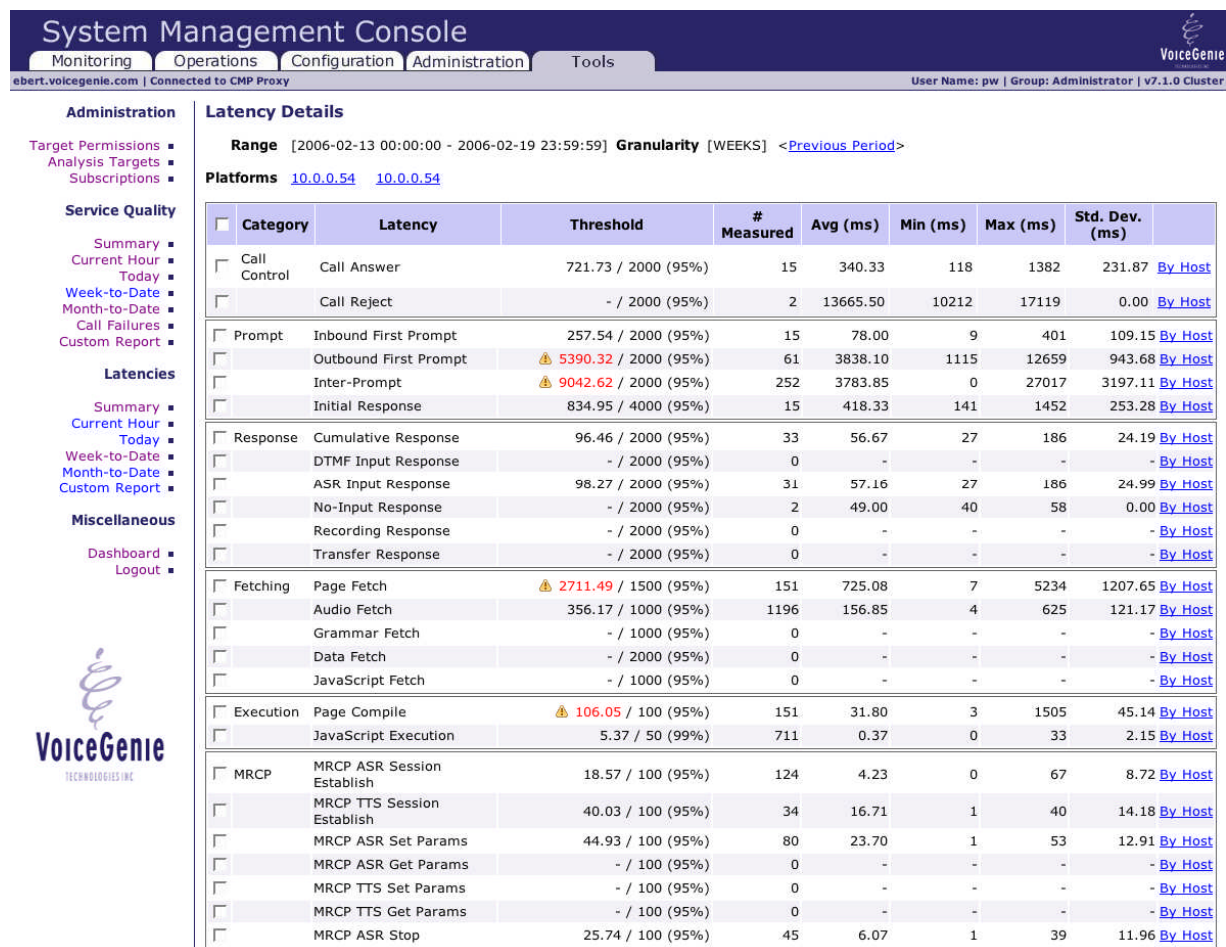
Figure 19: The Latency Details Screen

The following data is presented for each latency:

- `Category` – The category that the latency belongs to.

- `Latency` – The name of the latency described by this row of the table. The latencies are described in 2.3.4 System Latencies.

- `Threshold` – Indicates if the latency is meeting its configured threshold. Three numbers, namely a percentile value, a threshold value, and a latency value, are presented in the following format: `latency / threshold (percentile %)`. The latency and thresholds are in milliseconds. For example, if this field reads: `4000 / 5000 (95 %)`, then the threshold is configured such that the 95th percentile should be 5 seconds or less. The estimated 95th percentile latency value was calculated to be 4 seconds (It is important to realize that the percentile value is an approximation – see 5.3.2 Latency Thresholds). The text will be written in red and a warning icon will be displayed if the latency does not meet its configured threshold.

- `# Measured` – The number of latency measurements that were taken for all latency types in the category.
- `Avg (ms)` – The average latency value measured (in milliseconds). This average is calculated using all latency types in the category.
- `Min (ms)` – The minimum latency measured (in milliseconds). This minimum is calculated using all latency types in the category.
- `Max (ms)` – The longest latency measured (in milliseconds). This maximum is calculated using all latency types in the category.
- `Std. Dev. (ms)` – Represents the pooled standard deviation. This value is calculated using data from all latencies in the category. The standard deviation may be `0` in the following circumstances:
  - only one measurement has been obtained for the given latency
  - multiple measurements for the latency have been obtained, and each measurement has the same value (i.e. there is no observed variance)

The `By Host` link at the end of each row in the table shows detailed information on the latency broken down by host. See 8.13 The Latency Details By Host Screen for more details.

On the left-hand side of each row in the table there is a checkbox that can be used to select a latency for graphing. The checkbox in the top left corner of the table can be used to select or unselect all of the latency checkboxes. Selecting one or more latencies using the checkboxes and clicking on the `Generate Graph` button will create histograms of the selected latency values – see 8.16 The Latency Graph Screen for more details.

8.13 The Latency Details By Host Screen

This screen is similar to the `Latency Details` screen except that it shows latency information broken down by host. The table in this screen is similar to the table used in the `Latency Details` Screen (see 8.12 The Latency Details Screen) except that the `Category` column is replaced with a `Host` column. This column indicates the host to which the latency data applies. Each latency is listed for each host in the table.
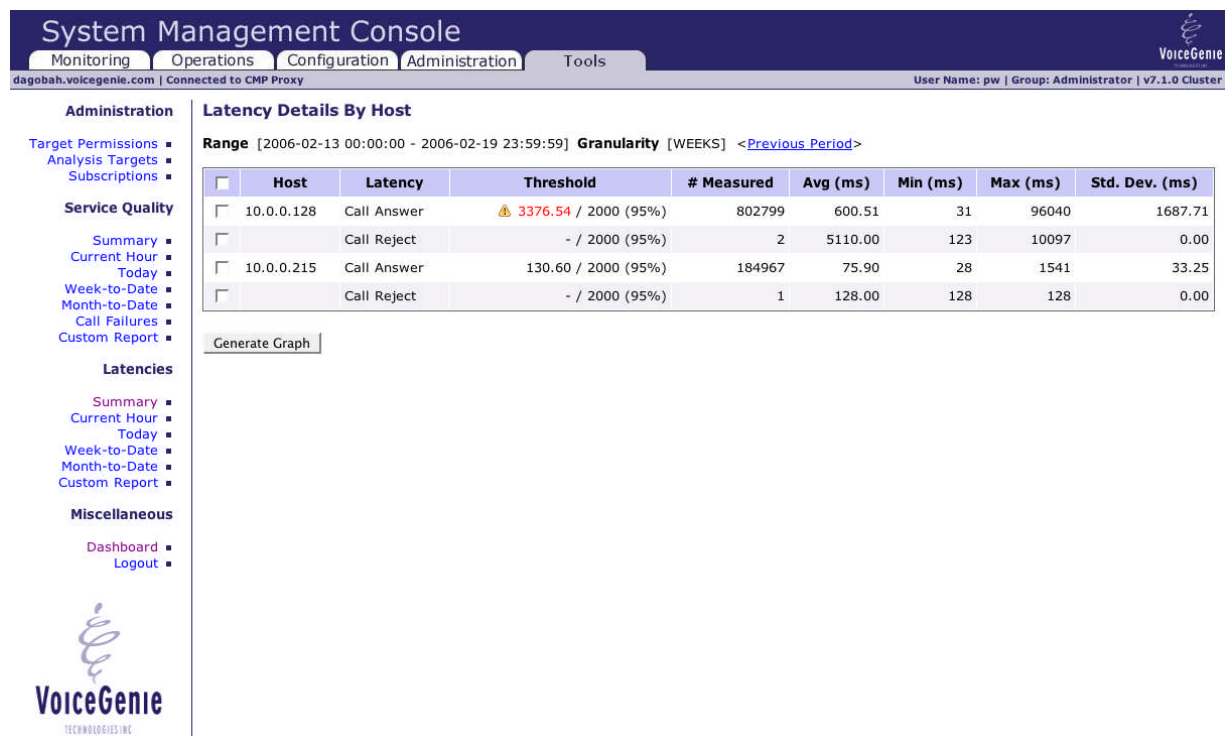
Figure 20: The Latency Details By Host Screen

On the left-hand side of each row in the table there is a checkbox that can be used to select a latency for graphing. The checkbox in the top left corner of the table can be used to select or unselect all of the latency checkboxes. Selecting one or more latencies using the checkboxes and clicking on the `Generate Graph` button will create histograms of the selected latency values – see 8.16 The Latency Graph Screen for more details.

8.14 The Custom Latency Report Request Screen

This screen is used to perform custom queries for latency reports. The user is presented with a set of fields and options that can be used to parameterize a query for detailed latency information.

Figure 21: The Custom Latency Report Request Screen

The available parameters are as follows:

- `From`, `To` – These parameters set the bounds for the time range for which latency information will be requested. Times must be entered in the following format: `YYYY-MM-DD hh:mm:ss`. The button to the right of the text fields can be used to select a time using a calendar widget.

- `Granularity` – This defines the time granularity for the latency data. Reports can be requested on an hourly, daily, weekly or monthly basis.

- `Latencies` – This parameter can be used to restrict the query results to one or more system latencies. Select `All latencies` to see results for all system latencies. If `Select latencies` is clicked, then a list of latencies will appear. One or more latencies can be selected from the list (Hold down the `Ctrl` key to select multiple latencies). If `Select latency category` is clicked, then a list of latencies categories will appear. One or more latency categories can be selected from the list; information for each of the latencies in the selected categories will be presented (Hold down the `Ctrl` key to select multiple latency categories).

- `Platforms` – This parameter can be used to restrict the query results to latency information gathered form specific platforms. Select `All platforms` to see results for all platforms. If `Select platforms` is clicked, then a list of platforms will appear. One or more platforms can be

selected from the list (Hold down the `Ctrl` key to select multiple platforms).

Once the query options have been selected, click `Submit` to perform the query. Results are displayed using the `Custom Latency Report` Screen described in 8.15 The Custom Latency Report Screen.

8.15 The Custom Latency Report Screen

This screen is used to view custom latency reports requested using the `Custom Latency Report Request` screen (see 8.14 The Custom Latency Report Request Screen). The information on this screen is nearly identical to the information in the `Latency Details By Host` screen (see 8.13 The Latency Details By Host Screen) except that this screen is able to display latency information for multiple time ranges. The results table contains an extra column labeled `Timestamp` that indicates the time to which the information in each row applies.

Figure 22: The Custom Latency Report Screen

On the left-hand side of each row in the table there is a checkbox that can be used to select a latency for graphing. The checkbox in the top left corner of the table can be used to select or unselect all of the latency checkboxes. Selecting one or more latencies using the checkboxes and clicking on the `Generate`

`Graph` button will create histograms of the selected latency values – see 8.16 The Latency Graph Screen for more details.

8.16 The Latency Graph Screen

This screen is used to display histograms of one or more system latencies. Each latency histogram specifies the system latency, and time range to which it applies. If the latency histogram applies to a specific platform, it is specified after the latency name in the title of the histogram. For example, if the histogram title is `Cumulative Response Latency,` then the histogram represents the Cumulative Response Latency for the entire deployment. If it is `Cumulative Response Latency @ 10.0.0.1,` then it represents the Cumulative Response Latency for the platform at 10.0.0.1.

Below the histogram title, the average, minimum and maximum latency values for the period are displayed. The standard deviation for the latency data is also displayed. The last line in the histogram heading indicates if the system latency is meeting its configured threshold criteria for the time period. Three numbers are presented: a percentile value, a threshold value, and a latency value. The latency and thresholds are in milliseconds. For example, if the last line reads: `Threshold 5000 ms, 95th percentile (est.) 4000ms,` then the threshold is configured so that the 95th percentile should be 5 seconds or less. The *estimated* 95th percentile latency value was calculated to be 4 seconds (It is important to realize that the percentile value is an approximation – see 2.3.5 Latency Thresholds).

The histogram graphically displays the distribution of latency values across a time range. The time range is broken down into segments that are displayed along the horizontal axis. The vertical bar that appears above each time range segment indicates the number of latency values that fall within that range. (The number of latency values will always be a whole number; however, because of limitations of the underlying graph library, the vertical scale may use floating point numbers in certain situations. While this does result in a graph that is somewhat harder to interpret, graph accuracy is not compromised.)

When no latency values fall within a time range segment, then no vertical bar is displayed. Holding the mouse pointer over one of the vertical bars in the histogram will display a tooltip that indicates the bounds of the time range segment and the number of latency values in that segment.
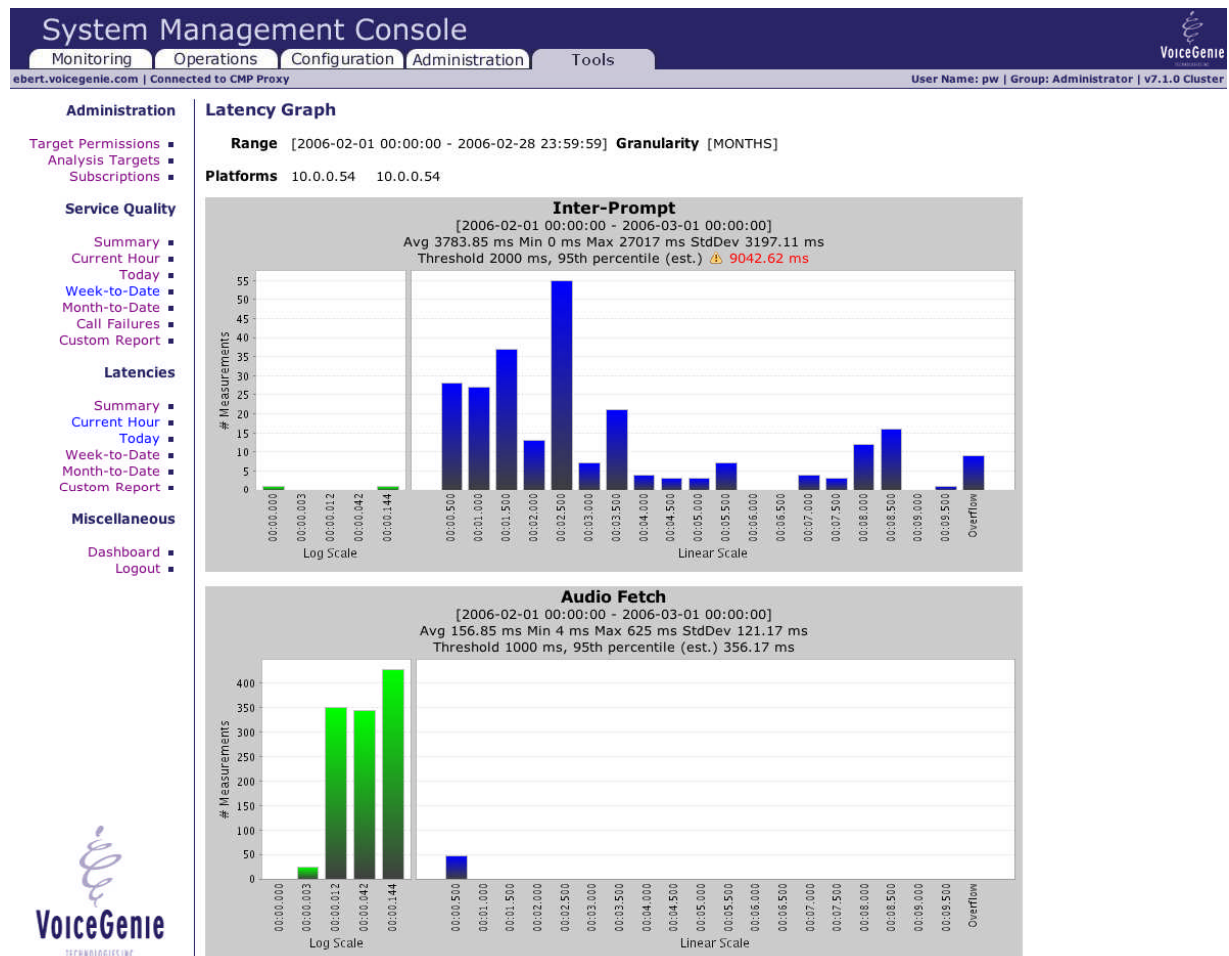
Figure 23: The Latency Graph Screen

The time range axis is split into two scales: a logarithmic scale and a linear scale. The logarithmic scale is used on the lower time values and the linear scale is used for the higher time values. This linear scale shows how the latency values are spread across the time range; the logarithmic scale breaks down and provides more details into the breakdown of the lower end of the time range.

Revision History

Version	Date	Change Summary	Author/Editor
Draft	April 21st, 2005	Initial release	Andrew Wahbe
1.0	May 4th, 2005	Incorporated changes from review	Andrew Wahbe
1.1	May 19th, 2005	Updated the "Configuring the Eclipse Client" Section with the new Call Analyst Preference Page included as part of RC3.	Derek Barnes
1.2	Sep. 14th, 2005	Added Quality Advisor Documentation	Andrew Wahbe
1.3	March 2nd, 2006	Updated for version 7.1	Albert Choi
1.4	Sep. 21st, 2007	Updated for version 7.2	Wen Wang