



Advanced Web Diagnostics Server

User Guide

Release 11.1

Please direct questions about Advanced Web Diagnostics Server or comments on this document to:

Technology Customer Support
Compuware Corporation
Customer Support Hotline
1-800-538-7822
FrontLine Support Web Site:
<http://frontline.compuware.com>

For telephone numbers in other geographies, see the list of worldwide offices at <http://www.compuware.com>.

Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation. Documentation may be reproduced by Licensee for internal use only. All copies are subject to the terms of this License Agreement. Licensee agrees to provide technical or procedural methods to prevent use of the Software and its documentation by anyone other than Licensee.

Copyright © 2009 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS—Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

ApplicationVantage, ClientVantage, Compuware, FrontLine, NetworkVantage, ServerVantage, Vantage, Vantage Analyzer, and VantageVieware trademarks or registered trademarks of Compuware Corporation.

Internet Explorer, Outlook, SQL Server, Windows, Windows Server, and Windows Vista are trademarks or registered trademarks of Microsoft Corporation.

Firefox is a trademark or registered trademark of Mozilla Foundation.

J2EE, Java, JRE, and Sun are trademarks or registered trademarks of Sun Microsystems.

Adobe® Reader® is a registered trademark of Adobe Systems Incorporated in the United States and/or other countries.

All other company and product names are trademarks or registered trademarks of their respective owners.

Build: October 16, 2009, 1:57

Contents

Introduction	5
Who Should Read This Guide	5
Organization of the Guide	5
Product Documentation Library	6
Customer Support and Online Information	7
Getting Help	7
Conventions	9
 Chapter 1 • Advanced Web Diagnostics Server Overview	11
AWDS Product Release Information	11
Supported Browsers and Connectivity	12
Internationalization Support	14
 Chapter 2 • Configuration Settings	17
Localizing the Report Server	17
Setting the Operating System to Display East Asian Languages	17
Localized User Settings	18
Changing Report Language at Login Time	18
 Chapter 3 • AWDS Standard Reports	21
Data Mining Reports	21
Report Tree	22
Advanced Web Diagnostics Overview Report	23
LOB Details Report	25
Application-URLs Reports	28
Slow Page Loads Reports	30
Slow Pages Classification	32
Slow Page Load Sequence Reports	36
Hit Details	36
Data Center Infrastructure Performance Reports	37
Network Performance Reports	38
Customer Care - Affected User Performance Reports	40
Continuous Improvement Gauge Report	42

Chapter 4 • Support for Oracle Forms	43
Chapter 5 • Basic HTTP Analysis	51
Orphaned Redirects Report	51
Portal Applications Report	52
User Path Through Site Report	52
Hit Details	55
Application Responses	56
Application Response Messages	57
Application Response Log	58
Chapter 6 • Transaction Monitoring	59
HTTP Transactions Overview Reports	59
Transaction Users and Agents Report	60
XML Transactions Overview Reports	61
Chapter 7 • Software Service Performance Reports	63
Slow Page Cause Breakdown	63
Load Sequence Stepchart	64
Query Log	64
Transaction Log	65
Operation Log	65
Chapter 8 • Alarms	67
Overview of the Alarm System	67
The Concept of Events, Alarms and Notifications	69
Types of Alarms	73
Means of Alarm Delivery	74
The Process of Defining an Alarm	74
Sending Alarm Notifications by E-mail	76
The Contents of Alarm Messages	78
Appendix A • Protocols Supported by AWDS	79
Appendix B • Dimensions and Metrics	81
Page Analysis data view	81
Transactions data view	100
Page Elements data view	108
Appendix C • Classification of Aborts	121
Index	123

Introduction

Who Should Read This Guide

This manual is intended for users of Advanced Web Diagnostics Server. It guides you through all of the features of Advanced Web Diagnostics Server and describes each top-level report and many lower-level reports, shows you how to interpret the reports, how to identify problems and how to optimize your network and site operation.

Organization of the Guide

This user guide is organized as follows:

- [Advanced Web Diagnostics Server Overview](#) [p. 11] – Gives a description of the Advanced Web Diagnostics Server product, product release information and a list of supported protocols.
- [Configuration Settings](#) [p. 17] – Describes configuration actions performed by individual report users and configuration settings related to an individual user, such as localization settings or password changes.
- [AWDS Standard Reports](#) [p. 21] – Describes the structure of the report tree and explains how to access different types of reports. Provides detailed descriptions of all of the standard reports provided with the product.
- [Support for Oracle Forms](#) [p. 43] – Introduces the specifics of Oracle Forms support.
- [Basic HTTP Analysis](#) [p. 51] – Describes the basic details of HTTP and HTTPS analysis.
- [Transaction Monitoring](#) [p. 59] – Provides the information about transaction monitoring reports and transaction users and agents.
- [Software Service Performance Reports](#) [p. 63] – Describes a number of reports that help you trace performance problems in software services, by letting you analyze details of page loads, query executions or transaction and operation executions.
- [Alarms](#) [p. 67] – Introduces the concept of an alarm and explains configuration for different methods of sending and receiving an alarm notification.

- [Protocols Supported by AWDS](#) [p. 79] – Provides a list of protocols that are supported by Advanced Web Diagnostics Server.
- [Dimensions and Metrics](#) [p. 81] – Gives a complete list of all the metrics and dimensions.
- [Classification of Aborts](#) [p. 121] – Describes classification of aborts, based on the transaction status.

Product Documentation Library

The following publications offer information on using and configuring Advanced Web Diagnostics Server.

ClientVantage Agentless Monitoring Release Notes

Summarizes new product features, known issues, and limitations, and lists last-minute information not included in other publications related to the product.

Distributed License Management – License Installation Guide

Describes how to install and administer Compuware product licensing components.

ClientVantage Agentless Monitoring Getting Started Guide

Introduces product components, release information, system requirements, licensing information, and performance estimates.

Advanced Web Diagnostics Server Installation Guide

Describes how to install the report server.

Vantage Agentless Monitoring Device Installation and Configuration Guide

Describes how to install the Agentless Monitoring Device, which collects data for the Vantage Analysis Server and Advanced Web Diagnostics Server.

ClientVantage Agentless Monitoring System Administration Guide

Describes how to configure and administer ClientVantage Agentless Monitoring.

Advanced Web Diagnostics Server on-line help

Provides on-line procedures and information to help you use the product.

Advanced Web Diagnostics Server User Guide

Guides you through the features of the report server. It describes each top-level report and many lower-level reports, shows you how to interpret the reports, how to identify problems and how to optimize your network and site operation.

ClientVantage Agentless Monitoring Web Services – Getting Started Guide for Developers

Provides data structure definitions and usage examples for CVAM Web service developers.

PDF files can be viewed with Adobe® Reader, version 7 or later. If you do not have the Reader application installed, you can download the setup file from the Adobe Web site at

<http://www.adobe.com/downloads/>.

Customer Support and Online Information

Corporate Web site

To access Compuware's site on the Web, go to <http://www.compuware.com>. The Compuware site provides a variety of product and support information.

FrontLine support Web site

You can access online customer support for Compuware products via our FrontLine support site at <http://frontline.compuware.com>. FrontLine provides fast access to critical information about your Compuware products. You can read or download documentation, frequently asked questions, and product fixes, or e-mail your questions or comments. The first time you access FrontLine, you are required to register and obtain a password. Registration is free.

Customer Support

You can contact Compuware Customer Support as follows:

- Web: via the “FrontLine Incident Reporting Form”.
- By phone: Compuware Customer Support.
 - USA and Canada customers: 1-800-538-7822 or 1-313-227-5444.
 - All other countries: please contact your local Compuware office.

All high-priority issues should be reported by phone.

Getting Help

When calling, please provide Customer Support with as much information as possible about your environment and the circumstances that led to the difficulty. You should be ready to provide:

- Client number: this number is assigned to you by Compuware and is recorded on your sales contract.
- The version number of the Agentless Monitoring Device (AMD) and the report servers.

For the report server

Use the report server GUI by selecting **Help** → **Product Information** → **About**, or **Tools** → **Diagnostics** → **System Status**.

For the AMD

Scroll down to the **Testing AMD** section. At the bottom of the diagnostic data paragraph, look for “Version ND-RTM v.ndw.x.yy.zz”.

- Environment information, such as the operating system and release (including service pack level) on which the product (AMD, report server) is installed, memory, hardware/network specifications, and the names and releases of other applications that were running.
Problem description, including screenshots.
- Exact error messages, if any (screenshots recommended).

- Whether or not the problem is reproducible. If yes, include a sequence of steps for problem recreation. If not, include a description of the actions taken before the problem occurred.
- A description of the actions that may have been taken to recover from the difficulty and their results.
- Debug information as follows:

Information from the report server

- Log files from `http://report_server_IP/root/log/` and `watchdog.log` from the `C:\Program Files\Common Files\Compuware\Watchdog` directory.
- Configuration file: `http://report_server_IP/ExportConfig`
- Screenshots of the problem.

Information from the AMD

Log files from `/var/log/adlex/`: `rtm.log`, `rtm.log.1`, `rtm_perf.log`, `rtm_perf.log.1`.

Information from the VCAEUE Server

- Log files from `..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE\cva\log` directory.
- All files from `..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE\platform3.0\InstallLogs`
- All `*.log` files from `..\Documents and Settings\All Users\Application Data\Compuware\<Service Name>\workspace\log\kernel` where `<Service Name>` is Microsoft Windows Service Name associated with VCAEUE Server. By default it is Agentless Platform 1
- Version file (`version.xml`) located in `..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE\`
- Version file (`version.xml`) located in `..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE\cva\eclipse`

Information from the VCAEUE Console

The installation log file:

`Vantage_Configuration_for_Agentless_End-User_Experience_11.1_InstallLog.log`
location:

`..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE`
log files located in the following directory of your VCAEUE Console installation:

`..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE\eclipse\log`
and version file (`version.xml`) located in `..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE\` and in `..\Program Files\Compuware\Vantage_Configuration_For_Agentless_EUE\cva\eclipse`.

NOTE

Please compress all the files before sending them to Customer Support.

Compuware values your comments and suggestions about the Vantage products and documentation. Your feedback is very important to us. If you have questions or suggestions for improvement, please let us know.

Conventions

The following font conventions are used throughout documentation:

This font	Indicates
Bold	Terms, commands, and references to names of screen controls and user interface elements.
Conventions [p. 9]	Links to Internet resources and linked references to titles in Compuware documentation.
Fixed width	Cited contents of text files, examples of code, command line inputs or system outputs. Also file and path names.
Fixed width bold	User input in console commands.
<i>Fixed width italic</i>	Place holders for values of strings, for example as in the command: cd <i>directory_name</i>
Menu → Item	Menu items.

CHAPTER 1

Advanced Web Diagnostics Server Overview

Advanced Web Diagnostics Server (AWDS) performs detailed HTTP analysis and delivers definite answers to customer problems, regarding Web site performance and errors. Faulty components can be quickly identified. Diagnosis can be extended behind the Web server, down to the particular Web application server.

AWDS is capable of transaction monitoring and can provide reports on the aggregation level, reflecting business processes instrumented by Web site applications. Performance, usage, and errors for each Web site user are mapped onto the business processes executed through Web site interaction.

AWDS can provide you with:

- Detailed HTTP analysis of every Web site user, including every single HTTP hit and page requested by the user, and recognition of individual Web forms.
- Transaction (predefined sequences of Web pages) monitoring with ready-to-use reports and drill-down capabilities.
- Problem-solving reports for Web sites that find systemic problems caused by HTTP-based application performance degradation.

AWDS Product Release Information

Table 1. Component module versions based on preceding and current release report server

Module name	Module version number in report server version		
	10.3.0	11.0.1	11.1.0
Advanced Web Diagnostics Server	10.3.0	11.0.1	11.1.0
DMI	10.3.0	11.0.1	11.1.0
RTM Base System	10.3.0	11.0.1	11.1.0

Module name	Module version number in report server version		
	10.3.0	11.0.1	11.1.0
RTM GATE	10.3.0	11.0.1	11.1.0
ND Core Base System	10.3.0	11.0.1	11.1.0

Supported Browsers and Connectivity

ClientVantage Agentless Monitoring users can access report servers through browsers with support for cookies, Java VM, JavaScript, and CSS 2. Before you start using the report server, it may be necessary to adjust JavaScript and HTTP 1.1 settings in your browser.

Compuware recommends the following browsers:

- Microsoft Internet Explorer version 6.0 or later with JavaScript and HTTP 1.1 settings enabled.
Note that due to a different handling of the data within the HTML, Microsoft Internet Explorer may experience degradation in performance while viewing reports containing a large number of columns or reports containing a large number of tooltips.
- Mozilla Firefox version 1.5.0 or later, with JavaScript, cookie support, and HTTP 1.1 enabled.
- Other browsers with support for cookies, Java VM, JavaScript and CSS 2 may also be used, but they are not recommended.

NOTE

- Some configuration screens require a Web browser with Java™ plug-in version 1.5.0.9 or higher.
- In Java plug-in version 1.5, TLS is turned off by default. This may cause some applets not to work in your Web browser. You must turn on TLS in the Java 1.5 Control Panel to have full access to all report server features. For more information, see [How to enable TLS 1.0 for Java 1.5 plug-in](#) [p. 13].
- Without JavaScript enabled, the top menu of the report server will not be visible and you will see the following message instead: "This product uses JavaScript. Please make sure JavaScript is enabled in your browser settings."

The Advanced Web Diagnostics Server and Vantage Analysis Server can be accessed using HTTP or, over secured connections, using HTTPS. We recommend secure access with a browser that supports TLS v.1. Using older versions of the protocol, such as SSL ver. 2 or SSL ver. 3, is not recommended but can be configured. For more information, see *Configuring the Report Server to Communicate over HTTPS* in the *Vantage Analysis Server – Installation Guide*.

How to enable JavaScript and support for HTTP 1.1 in your browser

Internet Explorer

To enable JavaScript:

1. Select **Tools** → **Internet Options** from the top menu in your browser and click the **Security** tab.
2. Choose the **Custom level...** button and enable **Active scripting** on the list of options.

To enable the HTTP 1.1:

1. Navigate to **Tools** → **Internet Options** and click the **Advanced** tab.
2. Scroll within the **Settings** list to the section titled **HTTP 1.1 settings** and make sure that the **Use HTTP 1.1** check box is selected.
3. Click **OK** and restart your browser.

Mozilla Firefox

To enable JavaScript:

1. Select **Tools** → **Options...** from the top menu in your browser and click the **Content** tab.
2. Select the **Enable JavaScript** check box.

To enable HTTP 1.1:

1. Open the browser and, in the address bar, type **about:config** and press [Enter].
The browser will display a list of current preferences.
2. Scroll to the **network.http.version** preference and make sure its value is 1.1. If the value is other than 1.1 it can be changed by double clicking on the parameter name.

How to enable TLS 1.0 for Java 1.5 plug-in

TLS for Java plug-ins is turned on in **Java Control Panel**, in the **Security** settings of the **Advanced** tab.

1. Access **Java Control Panel** in one of the following ways:
 - Windows control panel:
In Windows, click **Start** → **Settings** → **Control Panel** and select **Java** to open **Java Control Panel**. Note that **Java Control Panel** opens for the default Java installation whose number may be different than the plug-in's that you are trying to modify.
 - Java installation directory:
Navigate to the bin directory where the Java version you intend to modify is installed (for example C:\Program Files\Java\jre1.5.0_11\bin). Click the file `javacpl.exe` to activate the configuration tool.
 - Java platform icon in system tray:
Right-click the icon and choose **Open Control Panel** from the menu.
2. In **Java Control Panel**, click the **Advanced** tab and expand the **Security** tree.

3. Select the **Use TLS 1.0** check box.
4. Click **OK**.

Internationalization Support

Advanced Web Diagnostics Server supports international environments on both ends: report server and client browser.

Localized server support

The user interface of the report server is rendered in the following languages:

- English
- Japanese
- Korean
- Chinese simplified
- Chinese traditional.

For English, which is the default language setting, there is no need for additional configuration of the operating system or browser. To enable support for other languages, install the required font set for the target language and customize the regional options accordingly. For more information, see [Localizing the Report Server](#) [p. 17].

Character encoding support for monitored traffic

Advanced Web Diagnostics Server recognizes the following character encodings in monitored HTTP and XML traffic:

European:

- ISO-8859-1
- ISO-8859-2
- Unicode (UTF-8)

Japanese:

- Unicode (UTF-8)
- Shift_JIS
- EUC-JP

Korean:

- Unicode (UTF-8)
- EUC-KR
- ISO-2022-KR

Chinese:

- Unicode (UTF-8)

- GB18030
- Big5
- Big5-HKSCS
- EUC-TW
- ISO-2022-CN
- GB2312
- GBK
- HZ.

For more information, see *Character Encoding Support for Monitored Traffic* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

Configuration Settings

AWDS can work as a standalone application or as an application complimentary to VAS. In the latter case, apart from typical configuration tasks, such as configuring sites, areas, and regions, setting reporting groups, security options, or other report-related settings, you should also configure integration points with VAS.

Localizing the Report Server

The user interface of the report server is rendered in the following languages:

- English
- Japanese
- Korean
- Chinese simplified
- Chinese traditional.

The user interface and server reports are sensitive to the language settings of the browser, which are checked when you log in to the server and remembered throughout the session. Changing the language settings of the browser after you log in does not affect the current session. If your browser uses an unsupported language, English is used by default.

CAUTION

The report server will not work on systems where support for the English language is not installed.

Setting the Operating System to Display East Asian Languages

To display East Asian languages, change the Windows regional settings. The target language must be matched by the regional settings of the operating system where the report server is installed, and by the client system connecting to the server.

NOTE

East Asian characters are displayed correctly when using a Web browser run under a localized operating system. For example, Japanese characters are rendered correctly when reports or other screens are viewed using a browser on an operating system localized for Japan. Also, it is important that the report server generating the reports be localized for the language in which the reports are being viewed.

To enable support for East Asian languages:

1. In Windows, click **Start** → **Settings** → **Control Panel** and select **Regional and Language Options** to open the **Regional and Language Options** dialog box.
2. Click the **Languages** tab.
3. Select **Install files for East Asian languages**.
4. Click the **Regional Options** tab.
5. In **Standards and formats**, select a language.
6. *Optional:* In **Location**, select a geographical location.
7. Click **OK**.

Localized User Settings

Internationalization of the report server can be automatic, set by users per session, or set permanently by the administrator or users.

The report language can be set for each user individually.

- An administrator can set a user's language when adding a user to the server.
- A user can override the language setting at login time (see [Changing Report Language at Login Time](#) [p. 18]) or by modifying the setting in the configuration screen.

On the screens for adding or modifying users, you can choose a language from a menu or set the value to **Auto** to force automatic language detection. To override automatic language detection at the moment a new user is added or modified:

1. Click **Settings** → **Security** → **User Settings** to open the **User Settings** screen.
2. On the **Internationalization** panel, select a language (or **Auto** for language autodetection) from the **Language** menu.
3. Click **OK** to confirm your changes.
4. Log out and in again to ensure that the language setting has changed as intended.

The **Formats** panel displays detected regional settings such as the time and date format and the time zone of the server and client. These change according to the language setting of the browser.

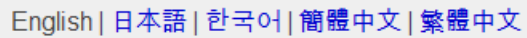
Changing Report Language at Login Time

At login time, you can change the report server language settings for a session.

Supported languages are listed at the bottom of the login screen. By default, the language setting is taken from the browser language settings, but you can switch the language for a new session by clicking a different language name before logging in.

The date and time display format, however, depend on the report server system locale unless the Web browser setting for the default language is different. For example, if a report server is installed and running on a Japanese Windows server, and a user with a Web browser localized for the Japanese language selects the English language for the reports at the login screen, all reports will be displayed in English, but dates and times will be displayed in Japanese. If that user instead sets the browser default language to Spanish, dates and times will be displayed in the Spanish language.

Figure 1. The list of supported languages displayed on the report server login screen

The image shows a horizontal list of language options separated by vertical bars. The options are: English, 日本語, 한국어, 簡體中文, and 繁體中文. The text is in a standard sans-serif font.

NOTE

East Asian characters are displayed correctly when using a Web browser run under a localized operating system. For example, Japanese characters are rendered correctly when reports or other screens are viewed using a browser on an operating system localized for Japan. Also, it is important that the report server generating the reports be localized for the language in which the reports are being viewed.

AWDS Standard Reports

AWDS is shipped with a set of reports that demonstrate how to use deep HTTP analysis data collected and calculated by AWDS. You need to import the report definitions onto the Dashboard page, since AWDS is intended to be used as an extension to VAS. In such a deployment, AWDS is transparently integrated with VAS, so you can use the full power of ClientVantage Agentless Monitoring.

Data Mining Reports

The dashboard page called **Data Mining Reports** is a collection of hyperlinks to useful reports defined by the user or provided with the product as example reports. By default, the example reports are not part of the initial installation and may be imported according to your needs. You can also add their saved reports to it.

To save a report on dashboard:

1. Select **Report** → **Load/Organize**.
2. Click the **Report Dashboard** tab.
3. Choose a report and specify the section of **Dashboard** the report will be placed in.

Remember that **Show the report...** must be selected for the report to be visible on **Dashboard**.

The main report sections on **Dashboard** are:

- Advanced Web Diagnostics ([Advanced Web Diagnostics Overview Report](#) [p. 23], [Continuous Improvement Gauge Report](#) [p. 42], [Customer Care - Affected User Performance Reports](#) [p. 40])
- Basic HTTP Analysis ([Orphaned Redirects Report](#) [p. 51], [Portal Applications Report](#) [p. 52], [User Path Through Site Report](#) [p. 52])
- HTTP Transaction Monitoring ([HTTP Transactions Overview Reports](#) [p. 59], [Transaction Users and Agents Report](#) [p. 60])
- XML Transaction Monitoring ([XML Transactions Overview Reports](#) [p. 61])

To add a report to a section that you define yourself, on the **Report Dashboard** tab, choose the **New** section from the list, and provide a name for it in the text field that appears on the right side of the list.

For more information on report import, see *Importing Report Definitions from External XML Files* in the *Data Mining Interface (DMI) – User Guide*.

Report Tree

Advanced Web Diagnostics Overview is a hierarchy of reports. The top-level report is accessed through the Advanced Web Diagnostics section of **Data Mining Reports**.

Figure 2. The **Data Mining Reports** link to the Advanced Web Diagnostics Overview report

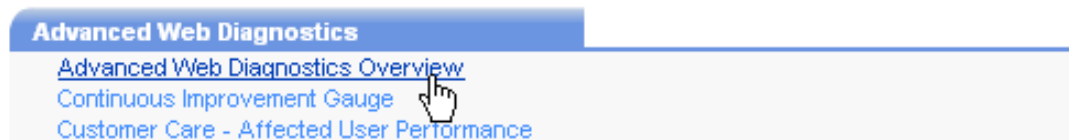
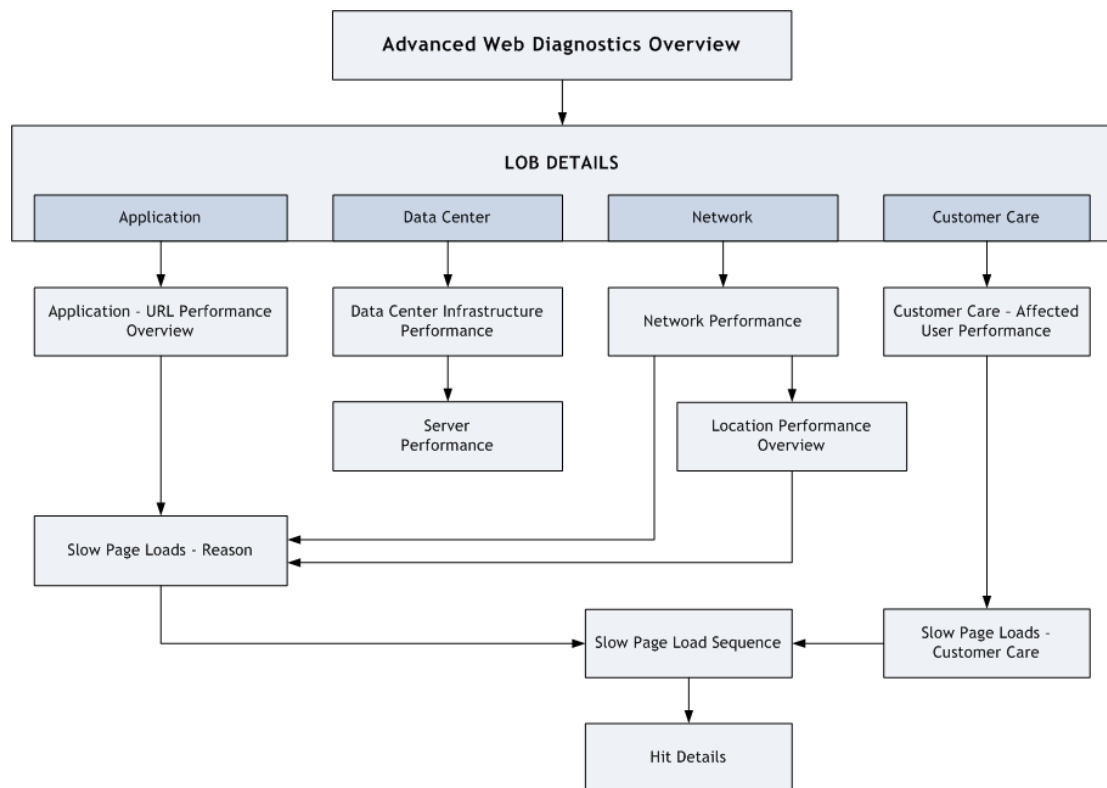


Figure 3. The **Advanced Web Diagnostics Overview** report tree



All of the Advanced Web Diagnostics Server tabular reports and graphs can be customized just like any other reports. For information on how to modify existing reports or create your own reports, please refer to the *Data Mining User Guide*.

To navigate between report pages, use the back and forward arrows provided next to the report titles. These arrows move through the history of the DMI reports you have visited. The standard

back one page and forward one page controls, supplied by your browser, could move between DMI reports only if no edits or filters were applied to the reports.

Advanced Web Diagnostics Overview Report

The **Advanced Web Diagnostics Overview** report is intended to be used by all the IT personnel responsible for technical support of users and for fault diagnosis and correction.

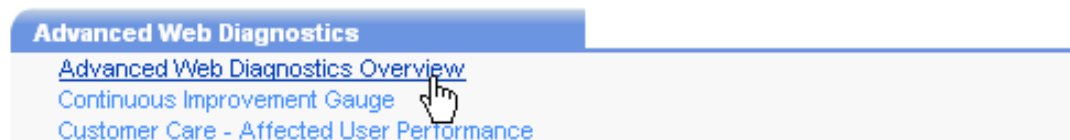
What can I learn from this report

The **Advanced Web Diagnostics Overview** report gives you an overview of the operation of all the reporting groups and allows you to focus immediately on those that require attention. It is a starting point for investigating performance problems.

For an explanation of the report dimensions and metrics, see [Page Analysis data view](#) [p. 81].

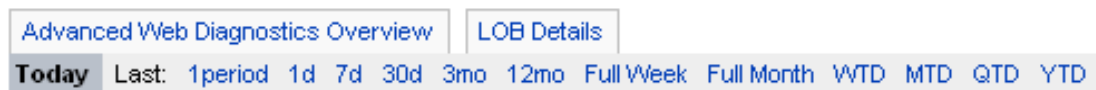
How to display the report

Clicking on the **Advanced Web Diagnostics Overview** link on **Data Mining Reports** brings up the **Advanced Web Diagnostics Overview** top-level report.



Many lower level reports—found lower down in the **Advanced Web Diagnostics Overview** report tree—also contain links to the **Advanced Web Diagnostics Overview** report. These links are provided as tabs above the time-range selection bar:

Figure 4. Links, from lower level reports, to the **Advanced Web Diagnostics Overview** and **LOB Details** reports

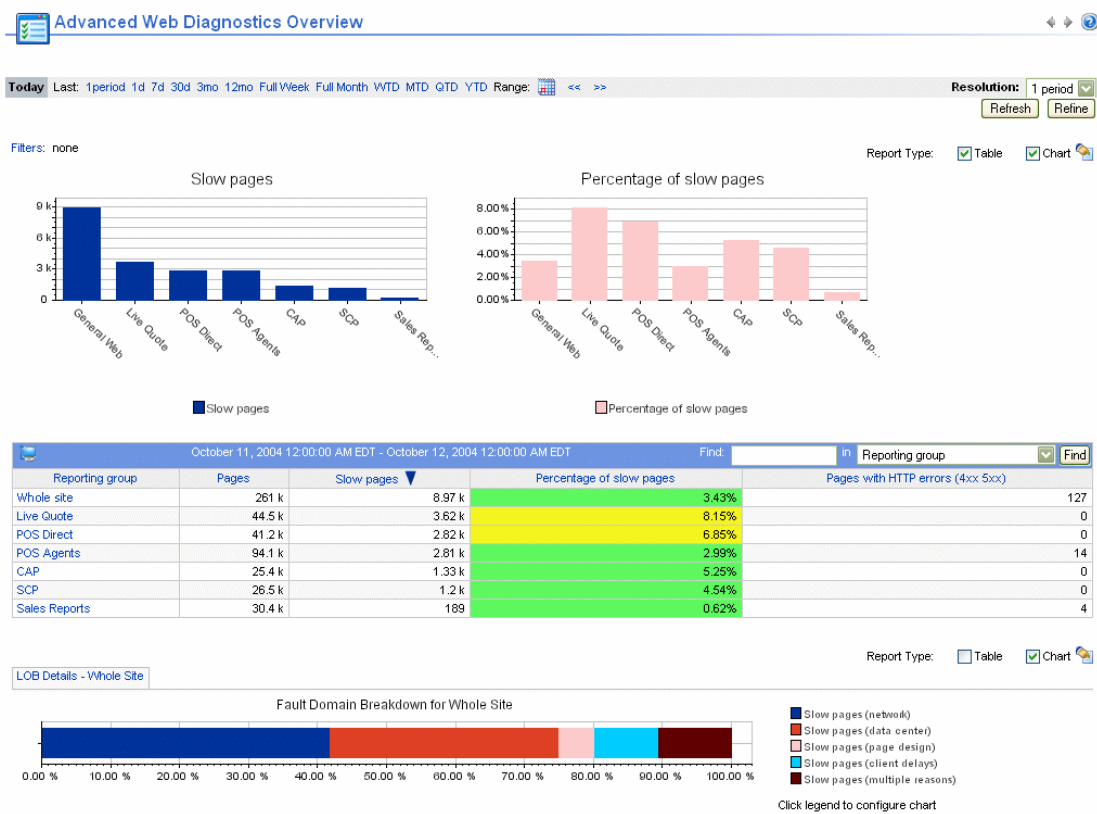


The following formats are used *regardless* of the detected language setting of the browser:

- Time values are displayed in “24-hour” format, with leading zeros.
- Dates are displayed in *short date* format. The language setting of the browser is used only to determine region-dependent details of the *short date* format.
- Year values are always displayed as 2 digits.

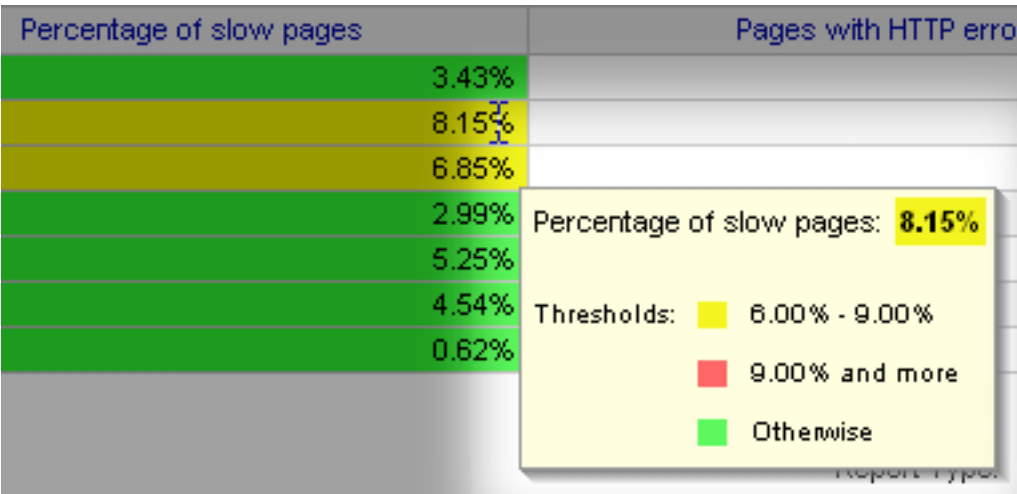
Report contents and usage

Figure 5. An example of the **Advanced Web Diagnostics Overview** report



The cell giving the percentage of slow pages is color-coded according to the legend given in the tooltip:

Figure 6. The tooltip showing the percentage of slow pages



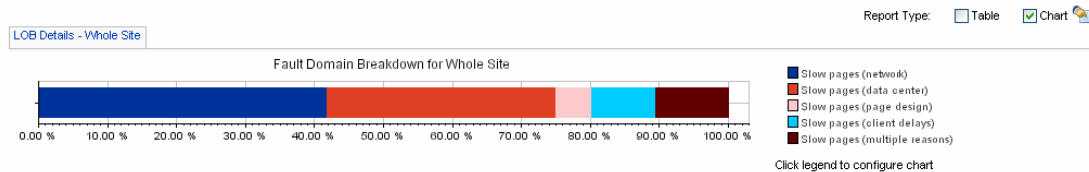
The number of slow pages and the percentage of slow pages are also shown as graphs.

NOTE

The order of reporting groups on the graphs follows that used in the report table, and can be changed by clicking on the appropriate report table column header.

Fault Domain Breakdown for Whole Site graph is given at the bottom of the report page and contains fault classifications for the entire Web site:

Figure 7. An example of the **Fault Domain Breakdown for Whole Site** chart

**Navigation**

Clicking the name of a reporting group invokes a LOB details report, for that particular reporting group.

Clicking the **LOB Details – Whole Site** tab, above the **Domain Breakdown** chart, activates the **LOB details for Whole Site** report.

LOB Details Report

The LOB Details reports are intended to be used by the IT personnel responsible *for a particular group of software services, servers or URLs in particular lines of business (LOB)*.

What can I learn from this report

The LOB Details reports give you an overview of the operation of a particular *reporting group*. They are starting points in investigating performance problems of the software services, servers or URLs in the reporting group.

How to display the report

Clicking the name of a reporting group in the Advanced Web Diagnostics Overview report, gives the LOB Details report for that line of business. Clicking the **LOB Details – Whole Site** tab, above the **Fault Domain Breakdown for Whole Site** graph, opens the **LOB Details for Whole Site** report.

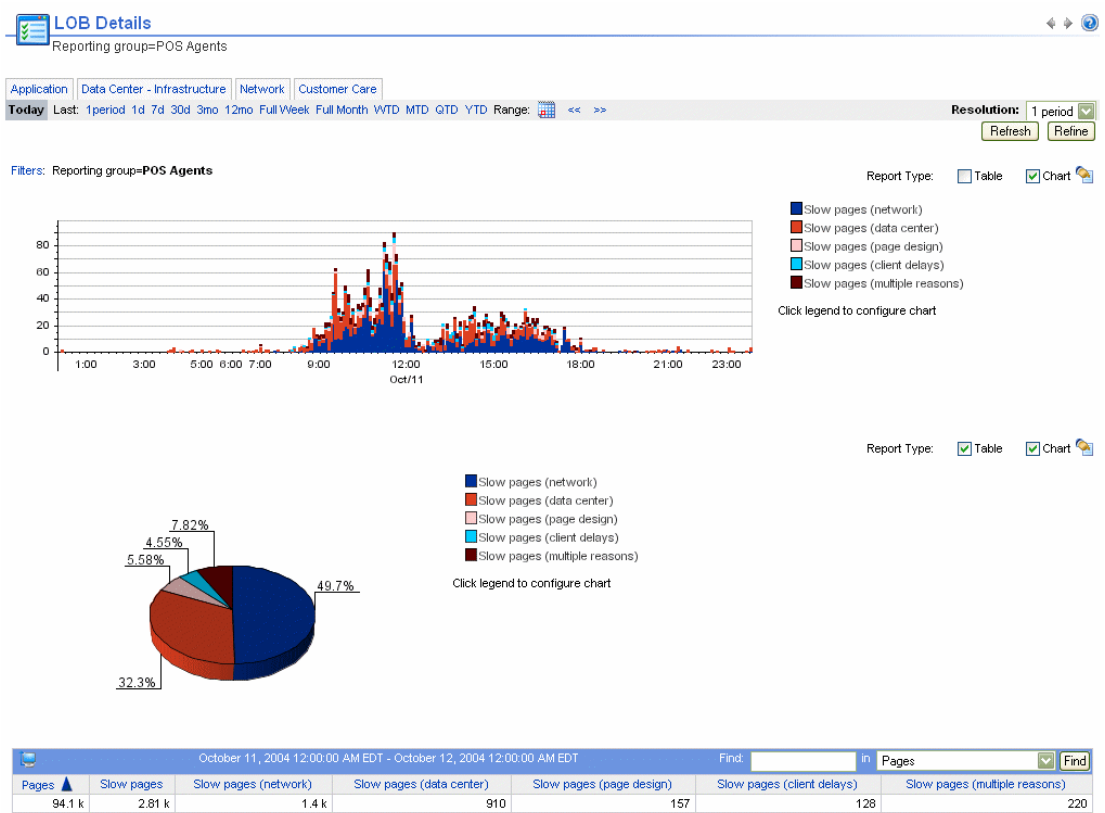
October 11, 2004 12:00:00 AM EDT - October 12, 2004		
Reporting group	Pages	Slow pages ▼
Whole site	261 k	8.97 k
Live Quote	44.5 k	3.62 k
POS Direct	41.2 k	2.82 k
POS Agents	94.1 k	2.81 k
CAP	25.4 k	1.33 k
SCP	26.5 k	1.2 k
Sales Reports	30.4 k	189

Many lower level reports—found lower down in the Advanced Web Diagnostics Overview report tree—also contain links to the LOB Details report. These links are provided as tabs above the time-range selection bar.

Figure 8. Links from lower level reports to the **Advanced Web Diagnostics Overview** and **LOB Details** reports



Report contents and usage



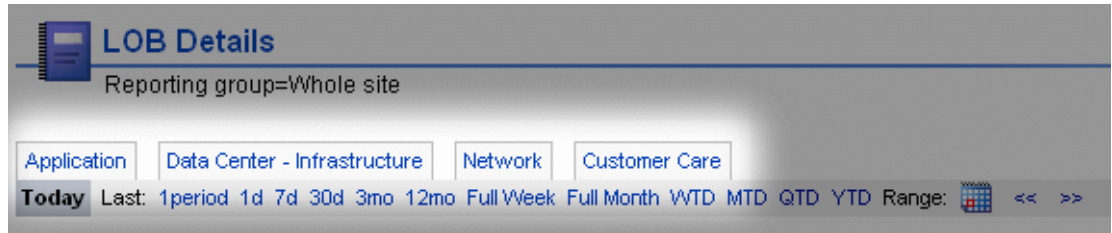
The report shows the performance and quality of the service, expressed as the number of slow pages due to the different reasons. For an explanation of the report dimensions and metrics, see [Page Analysis data view](#) [p. 81].

The information is presented in the following way:

- As a *graph* of slow pages against time is given in the upper portion of the report.
- As a *pie chart* giving the percentage distribution of slow pages, due to different reasons, is given further below.
- As a *table* giving the number of all pages and slow pages due to various reasons is provided at the bottom of the report page.

Navigation

The report is equipped with a number of tabs; each tab designed to deliver information tailored for that particular IT support group.

Figure 9. The LOB Details report navigation bar

Application-URLs Reports

These reports are intended to be used by an application software designer, to trace application performance problems for particular LOBs.

What can I learn from these reports

These reports are starting points for investigating problems caused by application software design, in particular LOBs. These could, for example, be problems caused by page size and the number of components on a page or problems caused by excessively time-consuming SQL queries.

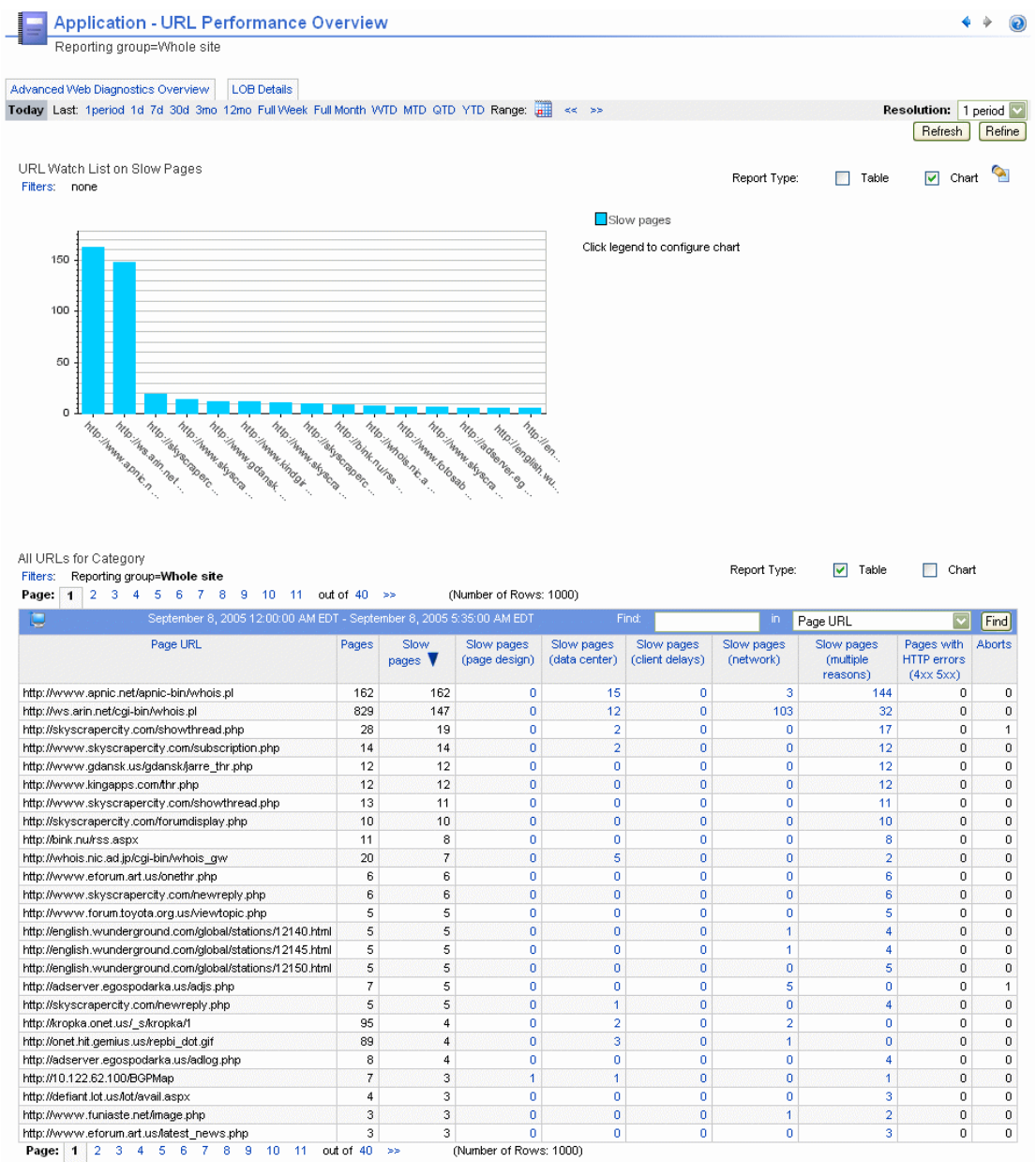
How to display the report

The report for a given LOB is activated by clicking on the **Application** tab, on the **LOB Details** report for that LOB.

The report for the whole site is activated by clicking on the **Application** tab on the **LOB Details for Whole Site** report.

Report contents and usage

Figure 10. An example of the Application - URLs report



The top of the report page shows the graph entitled URL watch list on slow pages. This graph shows 15 of the slowest URLs, that is URLs that have the greatest number of slow page loads. The graph helps you visualize how much the various URLs are slower than others, and helps you pick the slowest ones at a glance.

For an explanation of the report dimensions and metrics, see [Page Analysis data view](#) [p. 81].

Navigation

Clicking on the number of slow pages due to a particular reason, brings up a Slow Page Loads report for those specific URLs.

All of the reports that could be accessed through the links in the number of slow pages, in the **Application - URLs** report, are structured in a similar way, the only difference being the reason the slow pages occurred. Of particular interest to an application designer would normally be the number of slow pages due to *page design* and due to *data center*.

NOTE

The column giving the number of slow pages due to *page design* identifies those slow page loads that have been positively identified as being due to page design by, for example, being due to an excessive page size or a large number of page elements.

However, slow page loads apparently due to *data center* could also be caused by application design if, for example, page loads invoke time-consuming SQL queries. Similarly, the real reason behind slow page loads due to *client delays*, could also be because application design requires excessive processing on the client side.

For this reason, these additional metrics have also been included and the number of slow pages due to network has been provided for completeness.

Slow Page Loads Reports

These reports are intended to be used by the IT personnel to research software service, server or URL availability problems:

- *caused by a particular reason* or
- *experienced by a particular user*.

Some of these reports will be of interest to a number of different IT support groups. For example, slow page loads apparently due to *data center*, can be caused by server overload or application design, such as an excessive number of SQL queries. Thus both the server maintenance staff and application designers may need to make use of the same Slow Page Loads reports, though they may arrive at them from different points in the report hierarchy.

Similarly, slow page loads due to *client delays* are problems due to delays on client side (desktop delays). However, such delays can be, but do not have to be, due to the way pages are designed, thus requiring extensive processing on the client side or causing client side errors.

What can I learn from these reports

Depending on the context from which the reports were invoked, the reports show detailed information about individual slow page loads due to different reasons or for different users.

Statistics for the following reasons are provided:

- Page design
- Network

Network (Latency)

Network (Request Time)
 Network (Retransmissions)
 Network (Details Unknown)

- Data center delays
- Client delays
- Multiple reasons

NOTE

Slow pages due to *multiple reasons* are slow pages due to undetermined reasons. Note that the various categories of errors form non-intersecting sets, that is, the slow pages listed as being for multiple reasons do *NOT* appear in the statistics for individual reasons.

For information on the classification of slow pages, please refer to [Slow Pages Classification](#) [p. 32].

How to display these reports

The Slow Page Loads reports are generated through many different links in higher-level reports: On reports displaying the number of slow pages due to different reasons, clicking on a number of slow pages due to a particular reason, brings up a Slow Page Loads report for that specific reason, such as Page design, Data Center, Client Delays, Network, Multiple reasons.

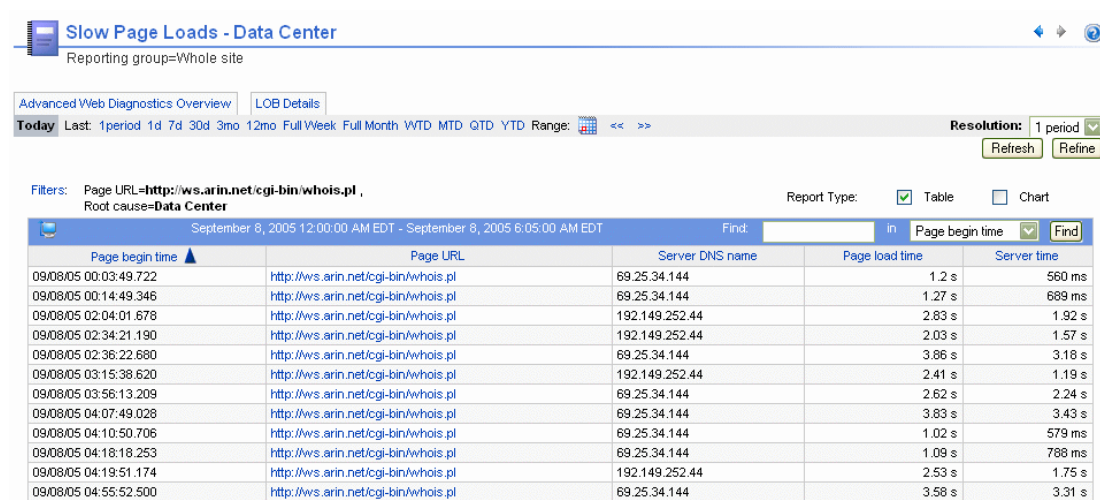
In addition, on the **Server Performance** report, clicking on the **Slow Page Loads - Data Center** tab, displays a **Slow Page Loads - Data Center** report.

Clicking on the user name on the **Customer Care - Affected User Performance** report brings up the **Slow Page Loads - Customer Care - Affected User Performance** report.

On the **Lob Details** report, clicking the **Network** tab brings up the **Network Performance** report.

Report contents and usage

Figure 11. An example of the **Slow Page Loads** report



By viewing higher level reports first—while performing a drill-down, on the way to a Slow Page Loads report—an IT support person would have narrowed down the problem to a particular number of URLs and a particular reason or a particular user.

Then, the **Slow Page Loads** report provides detailed information about particular instances of the slow page loads, allowing you to move the problem diagnosis further.

For example, having obtained detailed information about particular slow page load instances due to data center, an application designer could then view application server logs to determine server-related causes of the problem.

All of the Slow Page Loads reports contain a relevant list of URLs and a column giving page load begin times. Other information given in Slow Page Loads reports varies depending on the selected reason or context. Thus, for example, **Slow Page Loads - Page Design** reports show additionally *Root cause details*, *Page load time*, *Page size* and *Number of Hits*, while **Slow Page Loads - Customer Care** gives *Number of page errors*, *End-to-end RTT*, *Loss rate* and *Response throughput*.

Slow Pages Classification

Categories of slow pages

A page is classified as slow if it took longer time to load than its predefined threshold value, which by default is 8 seconds. A detailed classification of slow pages is based on the source problem that caused them to be slow: a *root cause*.

- Desktop
- Data Center
- Network

Network - Request Time

Network - Latency

Network - Retransmissions

Network - Other (Page Generation Time – Other and Throughput – Other)

- Page Design

Page Size

No. Components

- Other

Metrics involved in the classification of slow pages

The list of key metrics that are taken into account when classifying a slow page includes the following:

- Idle Time
- Page Generation Time
- Request Time
- Response Download Time
- Round-Trip Time
- Page Size
- Throughput

Interrelated with Round-Trip Time

Interrelated with Loss Rate

Interrelated with Hits

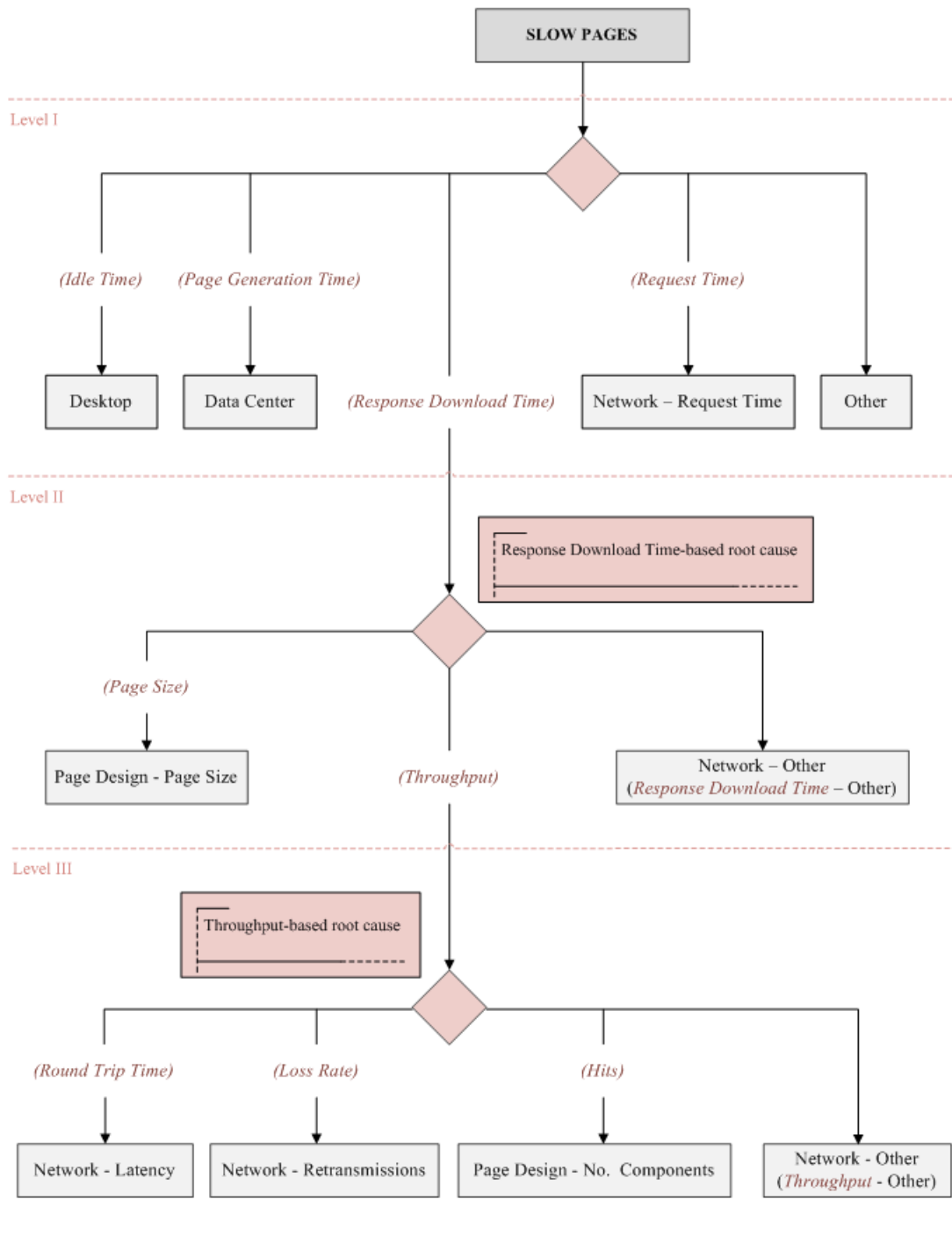
- Loss Rate
- Hits

Assigning a slow page to a specific category

Once a slow page is detected in the monitored traffic, a sequence of conditions is checked. The 3-level algorithm is based on the following criteria:

1. The actual value(s) of key metric(s) from the monitored traffic, for a given slow page.
2. The average value(s) of these metric(s), calculated for the last 20 calendar days, for a given URL.
3. Predefined thresholds.

If a given metric is proved to have significant influence on the degradation of Page Load Time (PLT) of a page, the page is assigned to a specific category.



Level I

- If *Idle Time* is responsible the degradation of *PLT*, the page is classified as **Desktop**.
- If *Page Generation Time* is causing the degradation of *PLT*, the page is classified as **Data Center**.
- If *Request Time* is causing the degradation of *PLT*, the page is classified as **Network – Request Time**.

- If *Response Download Time* is responsible for *PLT* degradation, we proceed to level II.
- The pages that could not be classified as any of the above, are assigned to a group named **Other**, which on the reports is represented as **Slow pages (multiple reasons)**. On the lower level reports, this category is divided into three subcategories based on the *root cause details*:
 - **Other**
 - **Other (no normal)** – *Normal* is the average value of a metric, calculated for the last 20 calendar days, for a given URL. The value is calculated only if the URL has been loaded at least 10 times. If the number of page loads is smaller than 10, the calculation is not made and the slow page is qualified as *Other (no normal)*.
 - **Other (faster than baseline)**

Level II

Pages with *Response Download Time* exceeding the threshold are then analyzed in terms of *Throughput* and *Page Size*, and checked against a threshold that defines the minimum size for a page to be considered slow because of the size, by default 100kB.

- If *Page Size* is the main cause of *PLT* degradation, and the page meets an additional condition defining the minimum size for a page to be considered slow because of the size at 100kB (by default), the page is classified as **Page Design – Page Size**.
- If the above condition is not met, and it is the *Throughput* metric that is responsible for the problem, the page is further analyzed at level III.
- The pages that could not be classified as either of these two, are assigned to **Network – Other**, which at this level is actually **Response Download Time – Other**.

Level III

At this stage, the slow pages with *Throughput* responsible for *PLT* degradation, are additionally analyzed in terms of *Round Trip Time*, *Loss Rate* and *Hits*.

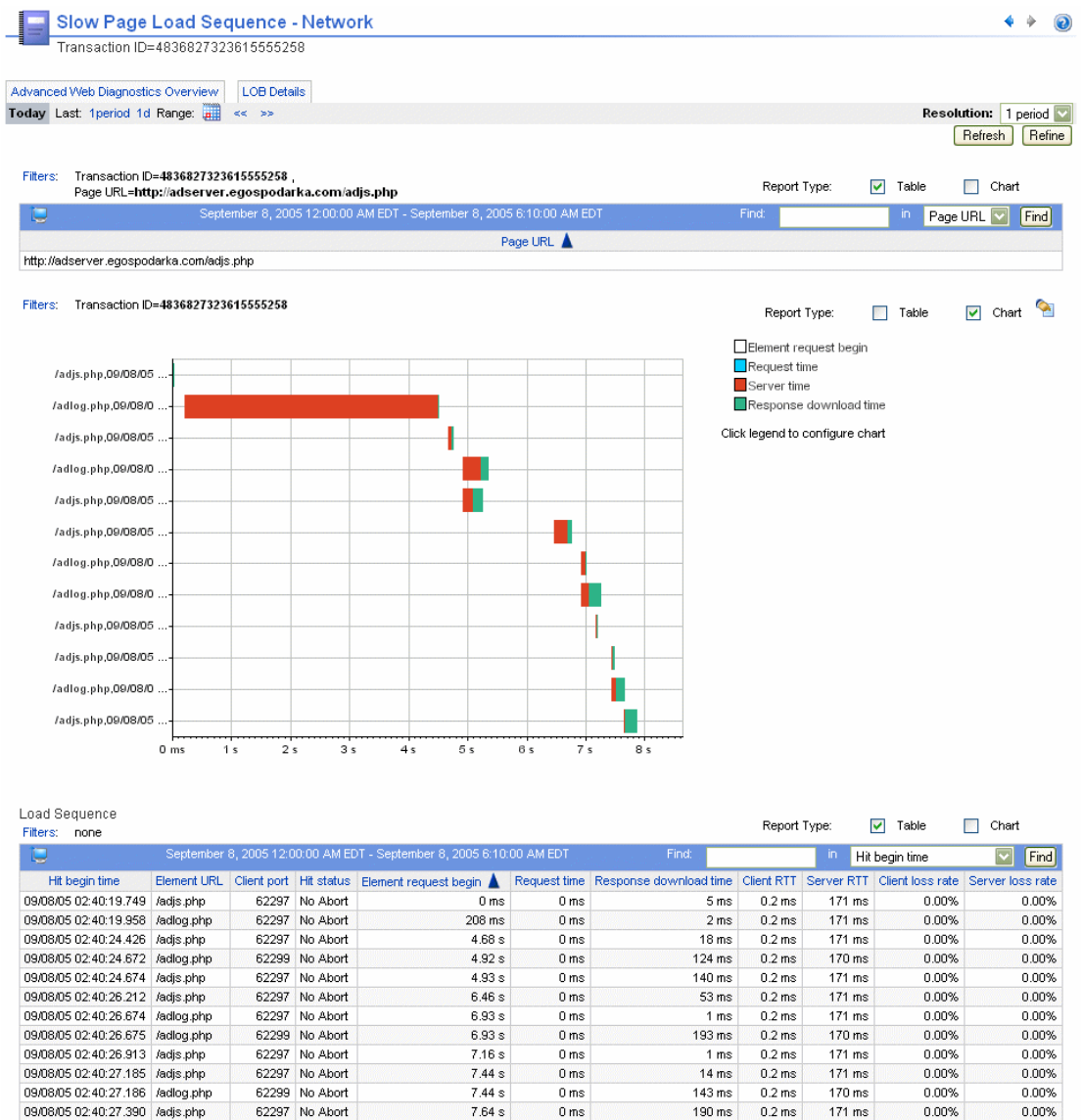
- If *Throughput related to RTT* is causing the deterioration of *PLT*, the slow page is classified as **Network – Latency**.
- If *Throughput related to Loss Rate* is the reason for degradation of *PLT*, a slow page can be classified as **Network – Retransmissions**.
- If *Throughput related to Hits* is the reason for degradation of *PLT*, and if the page meets an additional condition defining the minimum number of elements for a page to be considered slow because of the number of elements (by default 10), this page can be classified as **Page Size – No. Components**.
- And finally, the pages that could not be classified due to any of the above reasons, are assigned to **Network – Other**, which at this level is **Throughput – Other**.

Slow Page Load Sequence Reports

Clicking on a particular URL in the **Slow Page Loads** report table displays the **Slow Page Load Sequence** report for that particular page at that particular time.

The **Slow Page Load Sequence** reports are used in conjunction with the **Slow Page Loads** reports, to diagnose particular problems. Please refer to the description of the **Slow Page Loads** report for a discussion on the purpose and uses of these reports.

Figure 12. An example of the **Slow Page Load Sequence** report



Hit Details

This report is available from all reports presenting a page load process containing a link to the **Hit Details** report

Figure 13. Link to the Hit Details report

5/18/07 12:00:00 AM - 5/18/07 5:35:00 AM							
Component URL	Client port	Hit status	Component request begin	Request time	Server time	Response time	Response status
http://somecompany.booo.com/wp-content/uploads/2007/05/pimp-up-mask.jpg	4409	No Abort		0 ms	150 ms	17.3 s	

Filters: Transaction ID=0x464D64E2000A2ABA , Page URL=http://somecompany.booo.com/

Component URL: http://somecompany.booo.com/wp-content/uploads/2007/05/pimp-up-mask.jpg
Click to open Hit Details

It is a drill down report that represents an HTTP page hit, broken down into specific HTTP elements. By default, the table is presented in a transpose manner. This is beneficial when displaying elements like **Request header** and **Response header** making the table much more legible.

Figure 14. Sample Hit Details report

Hit Details		Component URL=http://baner.dictrs.com/adjs.php	
Today	Last: 1period 1h 1d Range: << >>	Resolution: 1 period	Refresh Refine
Report Type: <input checked="" type="checkbox"/> Table <input type="checkbox"/> Chart		Find:	Hit begin time Find
5/17/07 12:00:00 AM - 5/17/07 9:30:00 AM			
Hit begin time			
5/17/07 9:01:24.645 AM			
Request params			
n=653438463 what=zone:2 exclude=			
Request header			
GET http://baner.dictrs.com/adjs.php?n=653438463&what=zone:2&exclude=, HTTP/1.1 Host: baner.dictrs.com User-Agent: Mozilla/5.0+(Windows;+U;+Windows+NT+5.1;+en-US;+rv:1.8.0.11)+Gecko/20070312+Firefox/1.5.0.11 Accept: */* Accept-Language: en-us,en;q=0.5 Accept-Encoding: gzip,deflate Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7 Keep-Alive: 300 Proxy-Connection: keep-alive Referer: http://www.dictrs.com/plen?word=resentment&lang=EN			
POST data			
N/A			
Response header			
HTTP/1.0 200 OK Date: Thu, 17 May 2007 13:48:42 GMT Server: Apache X-Powered-By: PHP/4.3.11 Pragma: no-cache Cache-Control: private, max-age=0, no-cache Content-Length: 900 Content-Type: application/x-javascript Content-Language: en X-Cache: MISS from cnt-somesite.com X-Cache: MISS from cnt-somesite.com Proxy-Connection: keep-alive			

For explanation of an HTTP hit components, go to [Page Elements data view](#) [p. 108].

Data Center Infrastructure Performance Reports

These reports are of interest to server maintenance staff and are starting points for investigating problems caused, in particular LOBs, by data center infrastructure.

What can I learn from the reports

These reports provides an overview of problems and performance statistics of different servers. They can be used to diagnose problems caused by slow page generation due to server time, which could be caused by insufficient CPU power of the server, slow peripherals, insufficient

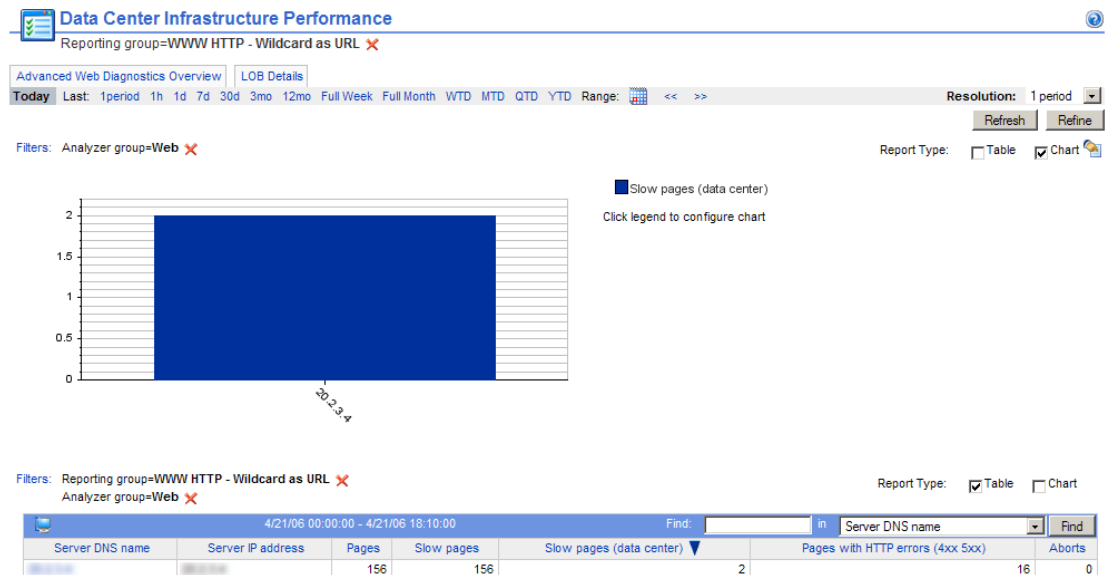
memory or swap space size, other operating system-related reasons or inappropriate load balancing between different servers.

How to display these reports

The report for a given LOB is accessed through the **Data Center – Infrastructure** tab on the **LOB Details** report.

Report contents and usage

Figure 15. An example of the **Data Center Infrastructure Performance** report



The top of the report page shows a graph of the number of slow pages due to data center. This graph shows 15 of the slowest servers, that is servers that had the greatest number of slow pages. The graph helps to visualize just how much the different servers are slower than others, and helps you to pick the slowest ones at a glance.

The table below the graph lists different servers, and for each server it gives the number of slow pages due to data center, the number of errors and the number of aborted transactions.

Network Performance Reports

These reports are of interest to network managers, responsible for maintaining network connections.

What can I learn from these reports

These reports are starting points for investigating problems, in particular LOBs, caused by network delays. In general, slow pages due to network can originate from two different reasons: problems in the Internet—not under the control of the monitored Web site—and problems with the connection of the Web site to the Internet.

How to display these reports

The **Network Performance** report, for a given LOB, are opened by clicking on the **Network** tab on the **LOB Details** report.

Report contents and usage

- the number and percentage of slow pages due to data center,
- the number and percentage of aborted transactions.

Figure 16. An example of the **Server Performance** report



The report consist of a table giving a more detailed breakdown of the number of slow pages due to different network-related reasons:

- loss rate
- latency (round-trip time)

- request time
- other network reasons

A graph of bandwidth usage and RTT is given at the bottom of the report page. This is useful for matching a rise in the number of slow pages due to latency, with an increase in the round-trip time, which confirms that the rise in the number of slow pages was caused by increased network latency.

Customer Care - Affected User Performance Reports

These reports can be used by Customer Care staff to verify application performance for specific users.

What can I learn from these reports

These reports list the users most affected by slow page loads. The reports can be used as starting points for investigating particular problems affecting particular users in the given LOB.

Note that the nature of the problem being investigated is not restricted at this point, since there are links embedded in the numbers of slow pages for all the possible reasons: page design, network, data center, client delays and multiple reasons.

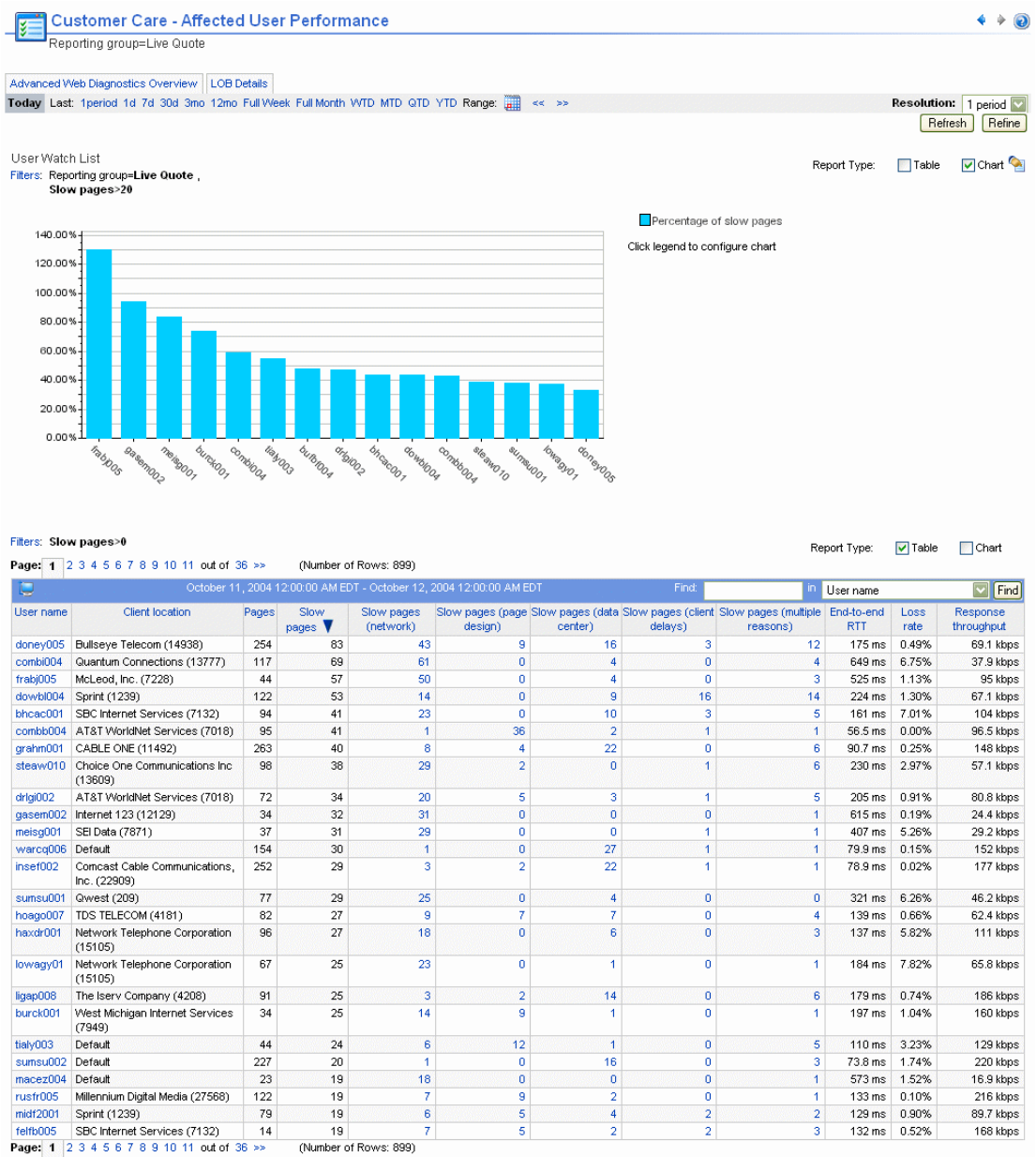
How to display these reports

The **Customer Care - Affected User Performance** report, for a particular LOB, is opened by clicking on the **Customer Care** tab, on the **LOB Details** report.

The **Customer Care - Affected User Performance** report for Whole Site is opened by selecting the report from **Data Mining Reports**.

Report contents and usage

Figure 17. An example of the Customer Care - Affected User Performance report



The report consists of a chart and a table. The chart shows the top 15 of the most affected users. The chart helps you visualize just how much different users are affected, and helps you pick the most affected ones at a glance.

The table gives a more detailed breakdown of the number of slow pages due to different reasons, as well as round trip time, retransmission and throughput information.

In order to find specific user information, enter user name in the **Find** box.

Clicking on a user name brings up a **Slow Page Loads - Customer Care** report. Clicking on the number of slow pages due to a particular reason, brings up the **Slow Page Loads** report for that reason.

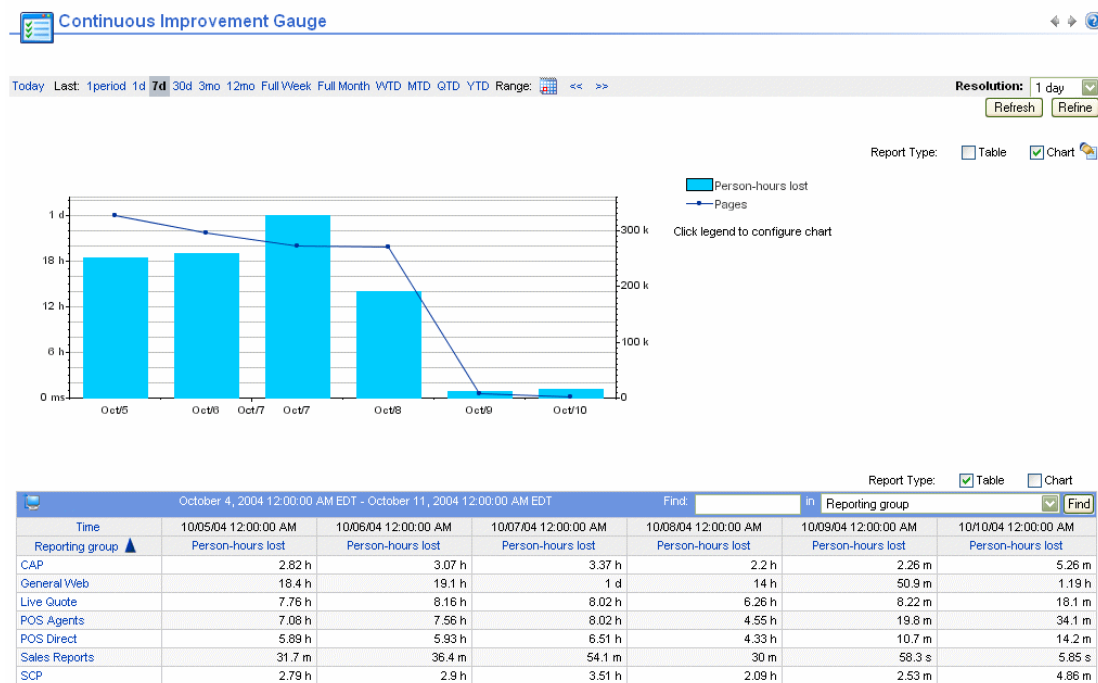
Continuous Improvement Gauge Report

This report gives an executive summary of the effects of long page load time on business.

The report is based on one metric, Person-hours lost due to performance, giving the total excessive time clients waited for pages to load. This is directly translatable into lost business revenue.

The table, provided in the lower portion of the report page, gives the information for each reporting group. Links in reporting group names bring up the **LOB Continuous Improvement Gauge** report for individual groups.

Figure 18. An example of the **Continuous Improvement Gauge** report



CHAPTER 4

Support for Oracle Forms

Using AWDS for monitoring Oracle Forms traffic requires special setup that switches HTTP-related metrics into the Oracle Forms meaning.

An AMD can be configured to monitor Oracle Forms traffic and pass on the traffic statistics to an installation of Advanced Web Diagnostics Server, where the data is displayed in all of the data views with the exception of Transactions. Oracle Forms statistics are not visible in the Transactions view and using Transaction configuration tools makes no sense for Oracle Forms traffic.

In such a configuration the meaning of AWDS report columns and DMI dimensions and metrics is changed to include aspects specific to Oracle Forms: metrics and dimensions related to HTTP-specific concepts should be, therefore, interpreted as relating to the corresponding Oracle Forms concepts. In particular, pages refer to Oracle Forms transactions, while the concept of hits is obsolete, in that the number of hits per page (per transaction) is always given as one. Those metrics and dimensions that require additional explanation, have been listed below on a per-view basis.

NOTE

To view Oracle Forms statistics you have to change the default **Web** filter setting for the **Analyzer group** dimension. Unless the filtering condition includes **Oracle Forms**, the Oracle Forms data will not appear on the AWDS reports. For more on filtering data refer to *Dimension Filter* in the *Data Mining Interface (DMI) – User Guide*.

Oracle Forms traffic analysis in the Page Analysis data view

Page Analysis dimensions—meaning specific to Oracle Forms

Page begin time

Oracle Forms transaction begin time

In addition to the time stamp of the measurement (the data package from the AMD), each transaction is separately stamped with its actual time of occurrence, with 1 millisecond

accuracy. This dimension can be used to list transactions one-by-one, in the order of their occurrence—to reflect exactly the client or server activity.

Page URL

Oracle Forms Transaction name

HTTP response status

not applicable

Page status

Oracle Forms transaction status: Abort or No Abort.

Root cause

not applicable

Root cause details

not applicable

Request method

not applicable

Page Analysis metrics—meaning specific to Oracle Forms

Page size

Transaction size: the number of bytes sent by the client and server during the transaction, without counting retransmitted bytes.

HTTP request time

Request time minus connection setup time.

Page load time

Transaction time

Server time

Server time of Oracle Forms transaction: the time it took the server to produce a response for the given request.

Note that this time is an aggregated time of server times for all data exchanges that were part of the Oracle Forms transaction. It is *not* the request-response time difference which is typical for other analyzed protocols.

Idle time

not applicable

HTTP server time

Equal to server time: the time it took the server to produce a response for the given request.

Note that this time is an aggregated time of server times for all data exchanges that were part of the Oracle Forms transaction. It is *not* the request-response time difference which is typical for other analyzed protocols.

Server think time

not applicable (always 0)

Hits

The number of Oracle Forms transactions

Hits per page

not applicable (always equal to 1)

HTTP errors and all metrics related to specific HTTP errors

not applicable

Percentage of slow pages due to DC

not applicable

Note that some data may appear for this metric. However, this data is calculated according to HTTP rules and the values rendered do not carry any meaning for Oracle Forms.

Transaction page begin

not applicable

Slow pages (*reason*)

not applicable - for any value of *reason*, such as **client delays** and other.

Note that some data may appear for these metrics. However, the data is calculated according to HTTP rules and the values rendered do not carry any meaning for Oracle Forms.

Number of request cookies

The number of key-value pairs in the Cookie field of the HTTP request header.

Request header size

Average length of the HTTP request header (sum for page hits).

Request body size

Size of the HTTP request body (sum for page hits).

Avg redirect time

not applicable

Number of response cookies

not applicable

Response cookie size

not applicable

Response header size

not applicable

Max number of request cookies

not applicable

Max request cookie size

not applicable

Max request header size

not applicable

Max request body size

not applicable

Max redirect time

not applicable

Max number of response cookies

not applicable

Max response cookie size

not applicable

Max response header size

not applicable

Min number of request cookies

not applicable

Min request cookie size

not applicable

Min request header size

not applicable

Min request body size

not applicable

Min redirect time

not applicable

Min number of response cookies

not applicable

Min response cookie size

not applicable

Min response header size

not applicable

Stdv number of request cookies

not applicable

Stdv request cookie size

not applicable

Std request header size

not applicable

Stdv request body size

not applicable

Redirect time

not applicable

Stdv number of response cookies

not applicable

Stdv response cookie size

not applicable

Std response header size

not applicable

Time resolution

not applicable

The remaining Page Analysis metrics retain their original meaning.

Oracle Forms traffic analysis in the Page Elements data view

Page Elements dimensions—meaning specific to Oracle Forms

Hit begin time

not applicable (equal to the Oracle Forms transaction begin time)

Request method

not applicable

Page URL

Oracle Forms Transaction name

Component URL

Oracle Forms Transaction name

HTTP response

not applicable

Hit status

Oracle Forms detailed transaction status.

Values:

- No Abort - no transaction abort
- Client abort - client aborted transaction
- Dead - Transaction error

Response status

not applicable (always OK)

Transaction status

not applicable (always “Belongs to page”)

Content type

not applicable

Remaining Page Elements dimensions retain their original meaning.

Page Elements metrics—meaning specific to Oracle Forms

Hits

The number of Oracle Forms transactions

Component request begin
not applicable (always 0)

Number of cookies
not applicable

Cookie total bytes
not applicable

Number of request cookies
The number of key-value pairs in the Cookie field of the HTTP request header.

Request header size
Average length of the HTTP request header (sum for page hits).

Request body size
Size of the HTTP request body (sum for page hits).

Avg redirect time
not applicable

Number of response cookies
not applicable

Response cookie size
not applicable

Response header size
not applicable

Max number of request cookies
not applicable

Max request cookie size
not applicable

Max request header size
not applicable

Max request body size
not applicable

Max redirect time
not applicable

Max number of response cookies
not applicable

Max response cookie size
not applicable

Max response header size
not applicable

Min number of request cookies
not applicable

Min request cookie size

not applicable

Min request header size

not applicable

Min request body size

not applicable

Min redirect time

not applicable

Min number of response cookies

not applicable

Min response cookie size

not applicable

Min response header size

not applicable

Stdv number of request cookies

not applicable

Stdv request cookie size

not applicable

Std request header size

not applicable

Stdv request body size

not applicable

Redirect time

not applicable

Stdv number of response cookies

not applicable

Stdv response cookie size

not applicable

Std response header size

not applicable

Time resolution

not applicable

Remaining Page Elements metrics retain their original meaning.

CHAPTER 5

Basic HTTP Analysis

A group of reports is designated to provide information based on basic HTTP analysis. The reports include Orphaned Redirects, Portal Applications, User Path Through Site, and Hit Details.

Orphaned Redirects Report

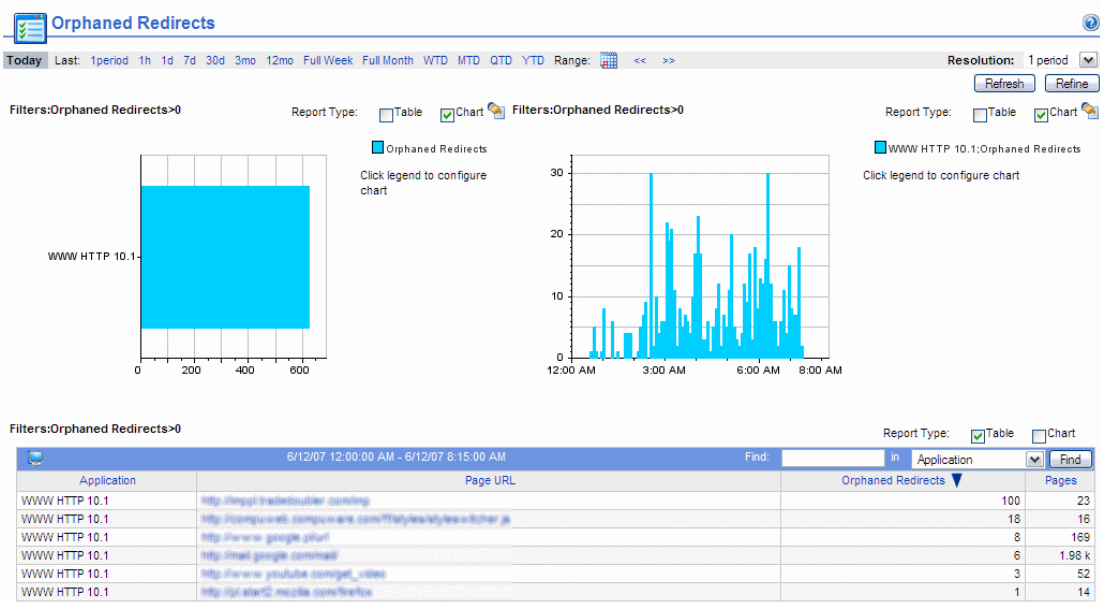
This report has been designed to show the orphaned redirects metric, for monitored applications that use the HTTP and HTTP SSL decrypted protocols. For each monitored application that uses HTTP or HTTPS, the report shows selected metrics, both in a graphical as well as tabular form. For an explanation of the metrics, see [Page Analysis data view](#) [p. 81].

While an orphaned redirect could occur for a number of reasons, the report can be used to show the number of redirects to sites which are not being monitored or are not visible and therefore appear as orphaned redirects.

NOTE

HTTP requests ending with a redirect response are stored until either a matching request to the target URL is seen (or at least a GET is seen), or a timeout expires. If the redirect target page has not been seen until redirect timeout expires, the AMD reports the URL with all transactional metrics equal to zero and the redirect request is referred to as an “orphaned redirect”. Such requests are always reported with the URL that caused the server to respond with a redirect status.

Figure 19. An example **Orphaned Redirects** report



Portal Applications Report

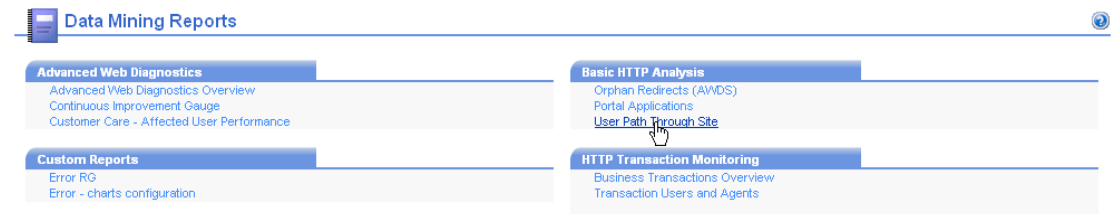
This standard report provides information on the performance of the monitored applications in the entire network in a selected period of time. It is accessed through the link in the **Basic HTTP Analysis** section on the **Data Mining Reports** screen.

This report can be used both for troubleshooting slow applications and for monitoring the performance; it provides information on the reasons for slow page loading. For explanation of report dimensions and metrics, go to [Page Analysis data view](#) [p. 81].

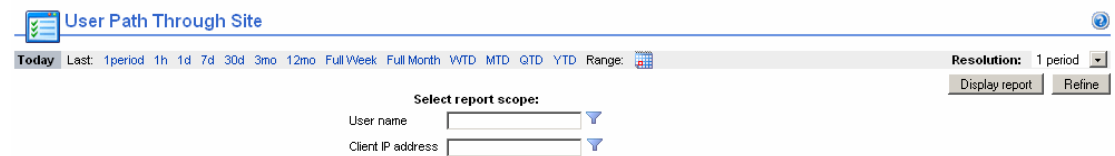
Each application can be investigated in detail after the link in **Page URL** column is clicked. Statistics for certain type of applications can be accessed when **Portal Applications - Groups** tab is clicked. Apart from the calculated values for the network and data center parameters the total time clients waited for pages to load in excess of the application threshold time is given (in the **Person-hours lost** column).

User Path Through Site Report

This standard report provides information on a user activity in a selected period of time, which by default is **Today**. It is accessed through the link in the **Basic HTTP Analysis** section on the **Data Mining Reports** screen.

Figure 20. Dashboard link to the Path Through Site report

The first thing you will see after clicking the link, is a form that enables you to define the report scope.

Figure 21. The scope definition form for a User Path Through Site report


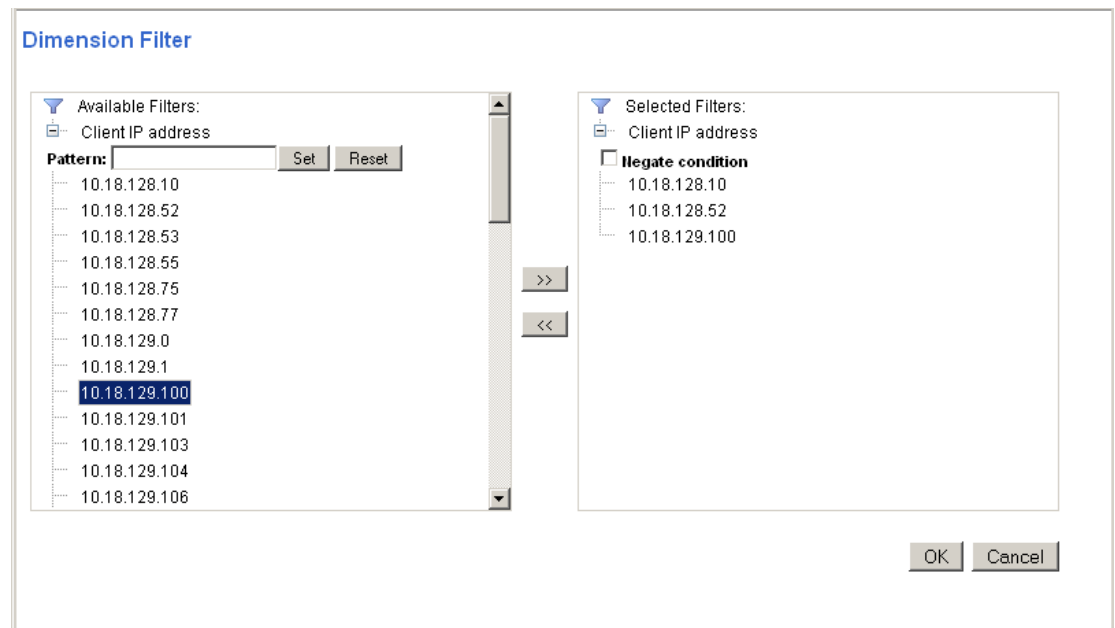
You can either input the parameters manually in the form, or use the  icon, which will open the **Dimension Filter** screen. It will enable you to select the user names and client IP addresses from the list in the **Available Filters** table and add them to the **Selected Filters**, using the >> and << controls.

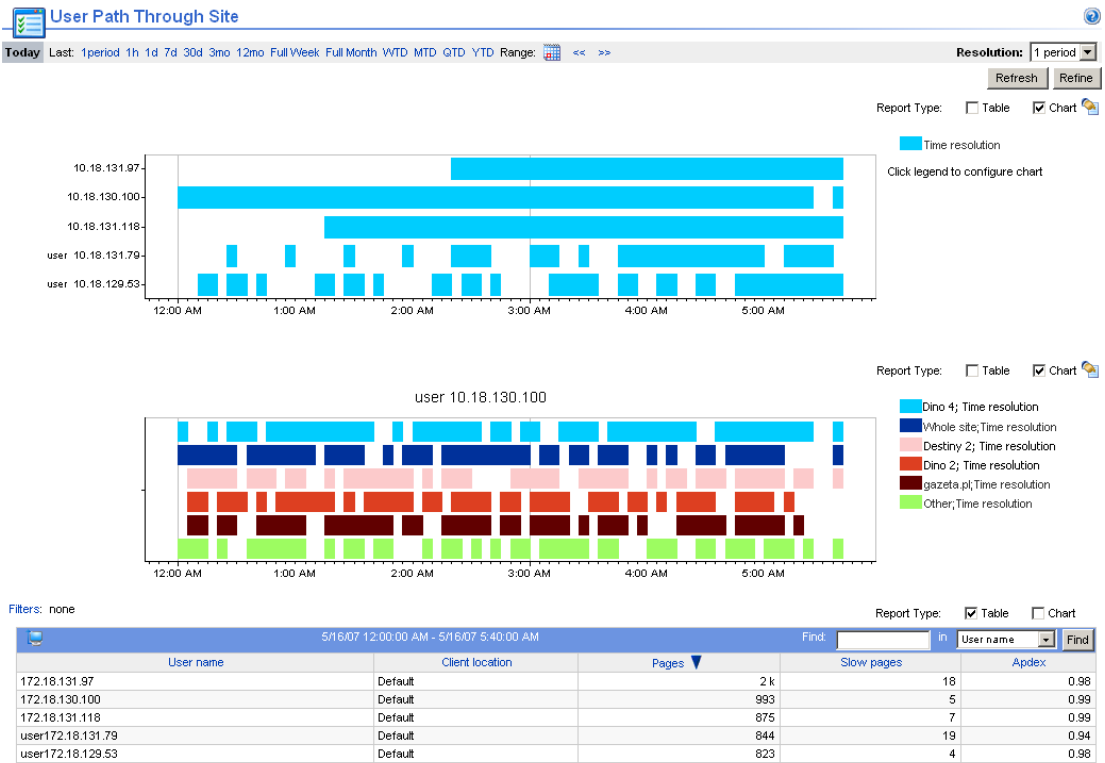
Figure 22. An example of the Dimension Filter screen with the selected filters

However, it is also possible to display the report without specifying the user name or IP address. The report will show results for the 5 most active users, either matching the given filter—if the report scope has been defined—or just 5 top results from among *all* the active users. By most active users we understand those with the greatest number of loaded pages.

The report consists of user activity charts and a summary table showing the statistics explained in [Page Analysis data view](#) [p. 81].

Different colors are used on charts to show the breakdown of users into reporting groups.

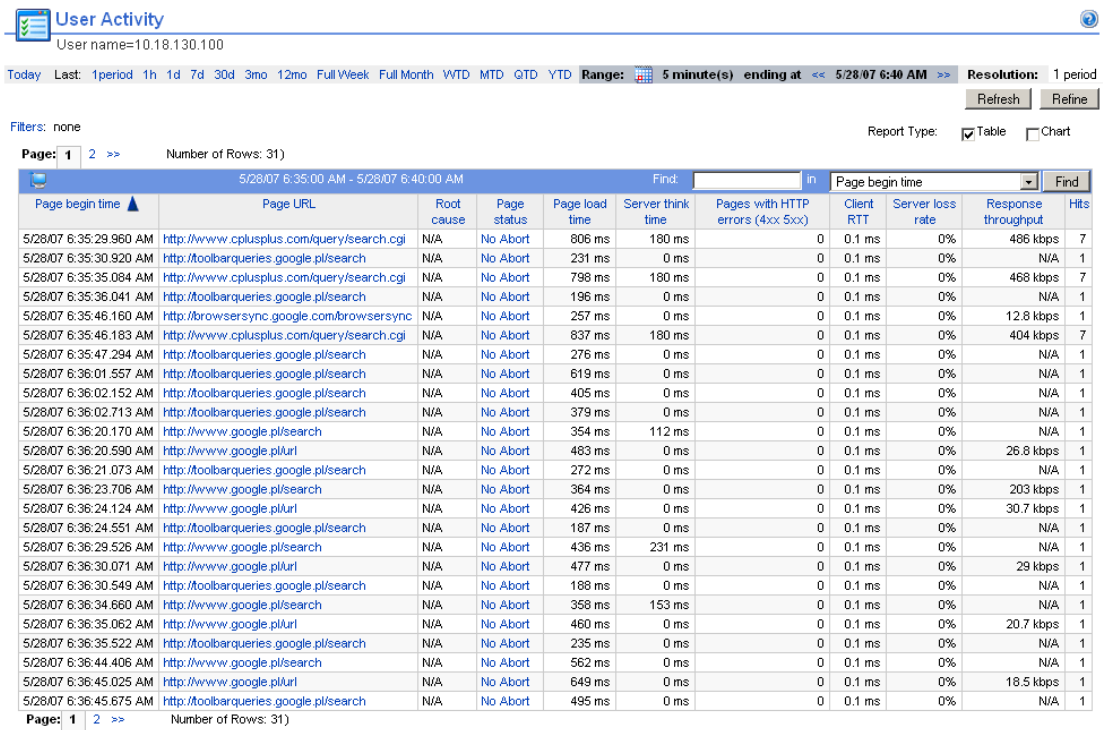
Figure 23. An example of User Path Through Site, top part



For each user name in the summary table for the report group and, for each default 5 minute activity period on the chart, the clickable bar leads to the **User Activity** report detailing the activity of the chosen time frame.

For each user name in the summary table, a default 5 minute activity periods are displayed on the chart. The report group chart combines the consecutive 5 minute intervals and presents it as one continuous activity period.

Figure 24. An example of the User Activity report

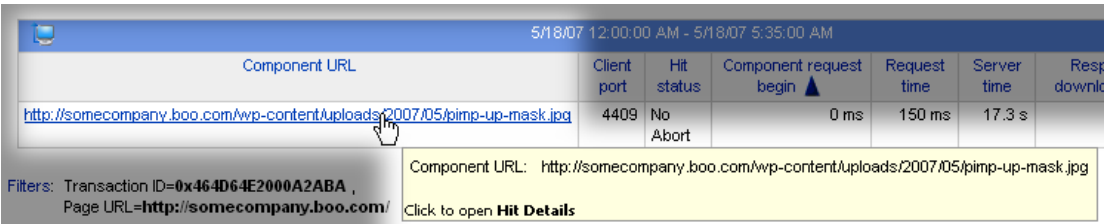


Using links nested in the **User Diagnostics** report, you can access the lower level reports, for example **Load Sequence-The Step Chart** and base **Hit Details** report, which is described in the [Hit Details](#) [p. 55] section of this manual.

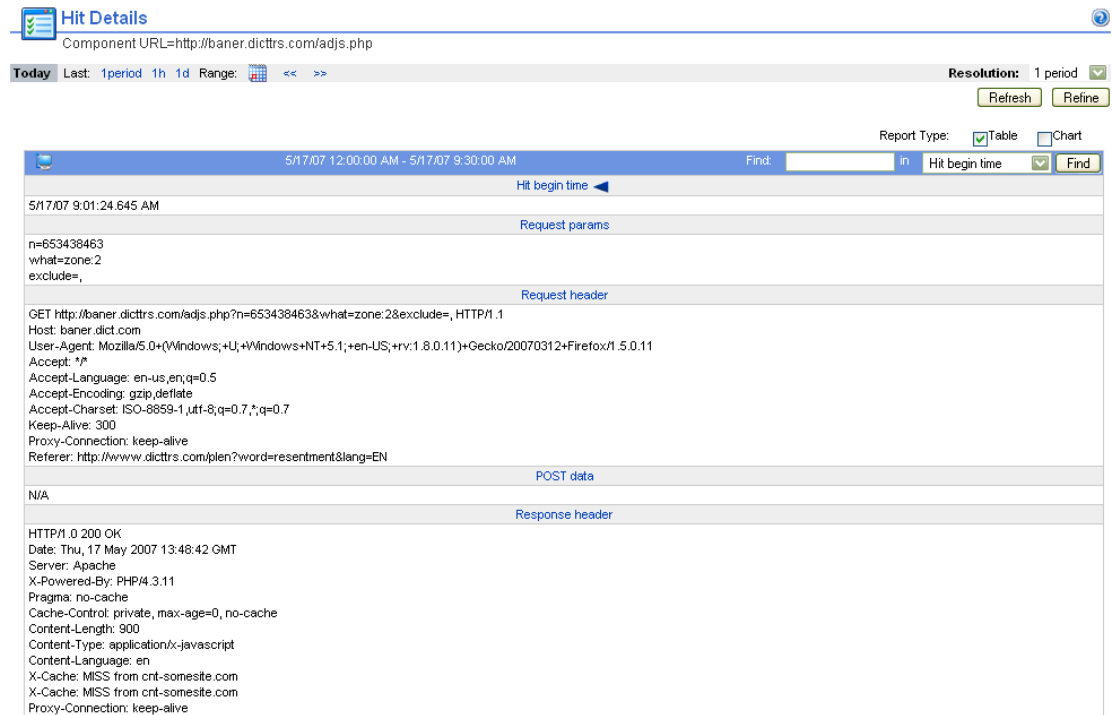
Hit Details

This report is available from all reports presenting a page load process containing a link to the **Hit Details** report

Figure 25. Link to the Hit Details report



It is a drill down report that represents an HTTP page hit, broken down into specific HTTP elements. By default, the table is presented in a transpose manner. This is beneficial when displaying elements like **Request header** and **Response header** making the table much more legible.

Figure 26. Sample Hit Details report

For explanation of an HTTP hit components, go to [Page Elements data view](#) [p. 108].

Application Responses

An application response is text communicated via HTTP page to a web application. Responses and response messages are reported if a given pattern is found in an HTTP data stream.

AMD analyzes and categorizes the messages returned by web application, enabling CVAM to diagnose which end-user actions cause specific responses or errors and to report on messages that end users have seen on their screens.

Pages delivered by web applications to clients contain various structured messages. Some, such as messages informing of the status of the transaction, are visible to the end user. Others are invisible and function as tags that are then used by the application client logic. CVAM can analyze the full content of an HTML response delivered by the server to the client, find specific patterns indicating such tags and messages, and retrieve their content. Use AMD to define and monitor HTTP application responses, such as errors that are not reported by means of standard HTTP error codes in HTTP headers, but that are communicated to the user by means of HTML pages with error messages. Such responses can also be reported if a given pattern is missing from an HTTP data stream. For more information, see *Extracting HTTP Application Responses* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

VAS counts all occurrences of a specific tag or pattern in the response and AWDS lists all values that matched the pattern. The auto-learning algorithm picks up all messages that match the pattern and uses them as a dimension for reporting. There is no limitation on the number of auto-learned messages.

At configuration time, application responses can be grouped into categories that act like virtual containers or response category identifiers. You can configure and rename up to five such categories globally. For more information, see *Defining Names of HTTP Application Responses and SSL Errors in Control Panel* in the *ClientVantage Agentless Monitoring – System Administration Guide* and *User-Defined Metric and Dimension Names* in the *Vantage Analysis Server – User Guide*.

Application Response Messages

For a selected software service, this report shows all response messages collected within a given response category (as configured on the AMD) or response messages for all response categories (summary view). It is accessed through the links in the **Application responses (X)** and **Application responses** values on the **End-User Experience** and **Website Status** reports.

By default, only the **Application Responses** column is enabled on the **Website: Status** reports, and the specific response categories (**Application responses (X)**) are hidden. On **End-User Experience** reports, the default report view does not include any application response information. To make hidden columns appear on the report, use the **Customize Columns** option in the report configuration menu. For more information, see *Customizing and Saving Tabular Reports* in the *Vantage Analysis Server – User Guide*.

Use the report to learn the number of times each response message occurs within a selected response category or within all response categories. The report shows the response message text, the number of occurrences of each message, and statistics such as time-related metrics and a user count (**Unique users** column).

NOTE

If you changed the default response category name, the columns will be named according to your configuration (both metrics can be renamed on VAS in the **Customized names configuration** dialog). For more information, see *User-Defined Metric and Dimension Names* in the *Vantage Analysis Server – User Guide*.

To filter the presented results, click the **Filters** link above the table to open the **Dimension Filter** screen.

- To include a dimension (member) value, select the desired value and click the >> button. This will copy the selected value to the right-hand side pane.
- To remove a dimension value from the report, select it on the right-hand side pane and click the << button. The selected item will be removed from the list of included filter criteria.

Figure 27. Example of the **Application response messages** report with filters active

Application response message (2)	Application responses (2)	Pages	Slow pages	Unique users	Page load time	Server time
Invalid or outdated authentication token	792	792	1.12M	567	198ms	131ms

Application Response Log

For a specific application response message, the **Application Response Log** shows details the exact time at which the message was reported, the page on which it occurred, and the name of the user to whom it was communicated.

To access the **Application Response Log**, click an application response message on the **Application Response Messages** report. Use this report to determine which end-user actions cause the server to communicate the specific response message you selected.

To filter the results by a specific user or URL, click the **Filters** link above the table to open the **Dimension Filter** screen.

- To include a dimension (member) value, select the desired value and click the >> button. This will copy the selected value to the right-hand side pane.
- To remove a dimension value from the report, select it on the right-hand side pane and click the << button. The selected item will be removed from the list of included filter criteria.

Figure 28. Example of the **Application Response Log** report

Page begin time	Page URL	User name	test 7 - Application responses (2)	Page load time	Server time
9/3/09 07:26:31.903	http://googleads.g.doubleclick.net/pagead/ads	10.18.129.34	1	304ms	234ms
9/3/09 07:26:34.083	http://googleads.g.doubleclick.net/pagead/ads	10.18.129.34	1	149ms	149ms
9/3/09 07:26:52.311	http://googleads.g.doubleclick.net/pagead/ads	10.18.129.34	1	157ms	77ms
9/3/09 07:26:56.197	http://googleads.g.doubleclick.net/pagead/ads	10.18.129.34	1	297ms	193ms
9/3/09 07:27:17.881	http://googleads.g.doubleclick.net/pagead/ads	10.18.129.34	1	216ms	158ms
9/3/09 07:28:11.915	http://googleads.g.doubleclick.net/pagead/ads	10.18.129.34	1	217ms	65ms
9/3/09 07:29:09.796	http://googleads.g.doubleclick.net/pagead/ads	10.18.129.34	1	46ms	43ms

Open in DMI

CHAPTER 6

Transaction Monitoring

This group of reports is aimed to analysis of transaction-related issues.

HTTP Transactions Overview Reports

The reports show statistics for the defined transactions, and help investigate performance, usage and errors for each monitored transaction.

You can access this group of reports from the **Transactions Overview** links on the **Reports → Data Mining Reports** screen, under **HTTP Transaction Monitoring**.

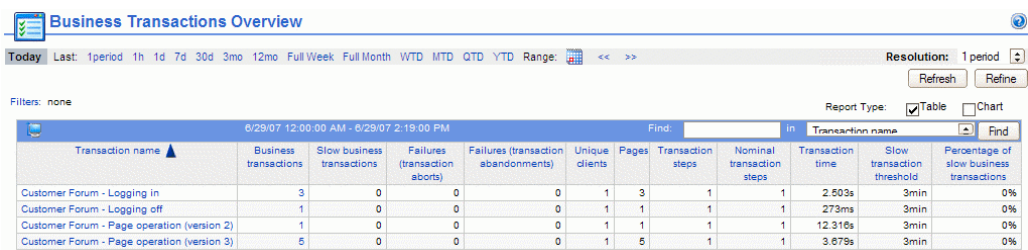
Depending on the resolution chosen, the reports show either raw or aggregated data. For resolutions equal to or greater than **1 day**, the reports are based on data aggregates: daily or monthly.

Monitored transactions are grouped by name. Each transaction can be investigated in detail by drilling down to a lower report level. For an explanation of report dimensions and metrics, see [Transactions data view](#) [p. 100].

The Transactions Overview group includes the following reports:

Transactions Overview (top level report)

Figure 29. An example of HTTP Business Transactions Overview report

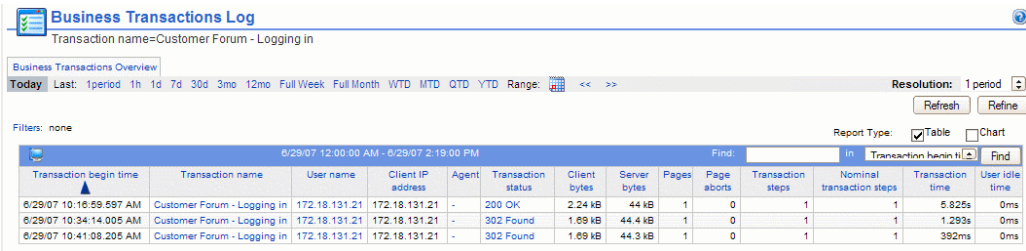


Transactions Log

Shows statistics for a particular transaction and is accessed by clicking a specific transaction name on the list of transactions on the **Transactions Overview** report . The

Business Transactions Log page shows details for separate instances of the selected transaction.

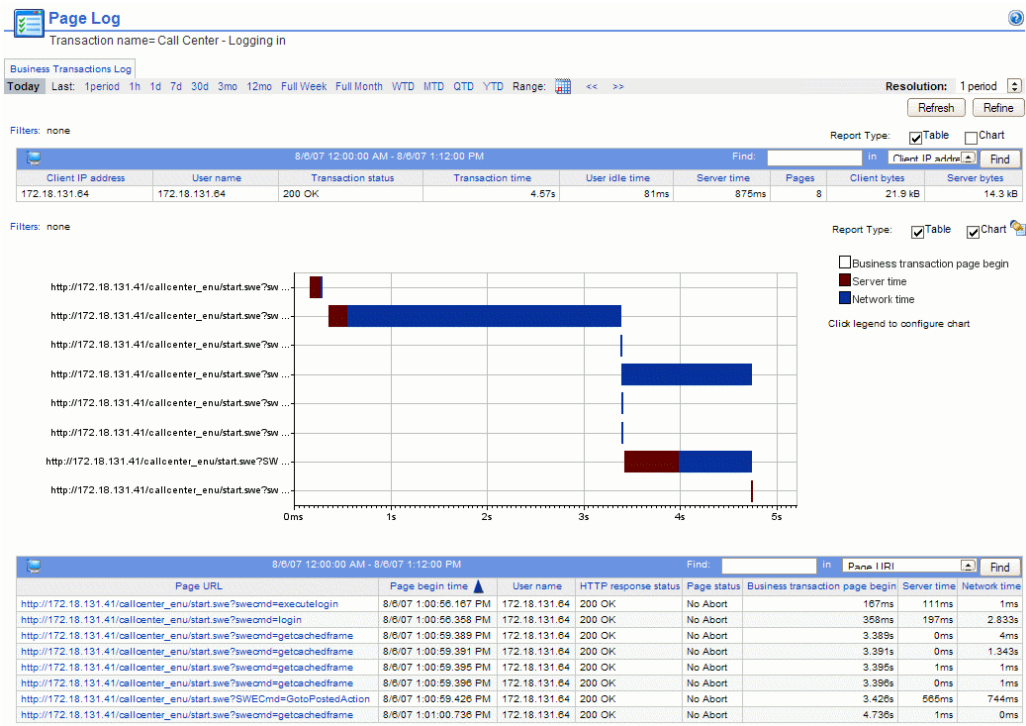
Figure 30. An example Business Transactions Log



Page Log

You can access the **Page Log** for a specific d HTTP transaction by clicking the **Transaction name** on the **Business Transactions Log** page. This level of the report allows you to view details for pages which create the sequence of URLs for the transaction.

Figure 31. An example Page Log for a defined HTTP transaction



Transaction Users and Agents Report

This standard report provides information on traffic generated by the real and synthetic users in a selected period of time, which by default is Today. It is accessed through the link in the **HTTP Transaction Monitoring** section on the **Data Mining Reports** screen.

This report helps to separate and also estimate the reliability of the traffic measurements done by Web analyzing robots, for example, Keynote, Gomez, Mercury. Two sets of data are presented

(tables or charts), which allow comparison between business transactions performed by real network users and monitoring agents.

The report presents user or agent names, IP addresses and sites together with the transactions statistics, explained in [Transactions data view](#) [p. 100].

The number of executed transactions is clickable showing the **Business Transactions Log** and then **Page Log**, described in [HTTP Transactions Overview Reports](#) [p. 59].

XML Transactions Overview Reports

Transactions Overview

You can access this group of reports from the **Transactions Overview** links on the **Reports** → **Data Mining Reports** screen, under **XML Transaction Monitoring**. The reports show statistics for the defined XML transactions, and help investigate performance, usage and errors for each monitored transaction.

Depending on the resolution chosen, the reports show either raw or aggregated data. For resolutions equal to or greater than **1 day**, the reports are based on data aggregates: daily or monthly.

Monitored transactions are grouped by name. Each transaction can be investigated in detail by drilling down to a lower report level. For an explanation of report dimensions and metrics, see [Transactions data view](#) [p. 100].

The Transactions Overview group includes the following reports:

- **Transactions Overview** (top level report)
- **Transactions Log**, which provides statistics for a particular transaction can be accessed after clicking a given transaction name on the list of transactions, on the Transactions Overview report.
- **Action Log**, allowing you to view details for pages which create the sequence of URLs for the transaction. You access it by clicking the **Transaction name** on the **Business Transactions Log** page.

Software Service Performance Reports

Software Service Performance Reports help you trace performance problems in software services, by letting you analyze details of page loads, query executions or transaction and operation executions.

Slow Page Cause Breakdown

The **Slow Page Cause Breakdown** shows slow page loads for a software service or a group of software services, classified according to various reasons and plotted against time, and shows the total numbers of slow pages and percentage breakdown for various reasons.

What can I learn from this report

The report is a starting point for investigating performance problems of software services and allows you to narrow down areas that need to be investigated, such as network, servers, page design and other.

Report contents and usage

The report shows performance and quality of service, expressed as the number of pages that were slow for various reasons. For an explanation of the report dimensions and metrics, see [Page Analysis data view](#) [p. 81].

The information is presented in the following ways:

- As a *graph* of slow pages against time.
- As a *pie chart* giving the percentage distribution of slow pages for various reasons.
- As a *table* giving the total number of pages and the total number of slow pages due to various reasons, all at the bottom of the report page.

Navigation

Click the number of slow pages due to a particular reason to go to the **Slow Page Loads** report for that reason. From there, click the page URL to display the **Slow Page Load Sequence** report

for that page. It gives you the actual stepchart for the load sequence and detailed values for the hit. From there, you can display the **Hit Details** report by clicking the entry in the **Component URL** column.

Load Sequence Stepchart

You can use the **Load Sequence Stepchart** report to examine time-related statistics for particular URL load sequences.

What I can learn from this report

The **Load Sequence Stepchart** report can be used in conjunction with **User Activity: User** and **Website Status: URLs** reports to trace performance problems with software services. The step chart and tabular data allow you to analyze page load details not available in VAS and to trace related problem such as slow page loads. Because it is possible to track page components through the load sequences to HTTP headers, you can also use it to complement to Web application debugging.

Report contents and usage

The load sequence is a combination of all individual page loads observed within a given interval. The report consists of a graph and two tables:

- Component URL and mean values common to all occurrences counted in the **Hits** column.
- A chronological list of all page loads whose URL components formed the step chart.

Navigation

From the table showing sequence elements, you can click the **Page Name** to drill down to details for individual page loads. From there, click the entry in the **Component URL** column to get to the **Hit Details** report, which gives you details of request/response headers.

Query Log

Use the **Query Log** report to examine time-related statistics for particular query execution sequences.

This report can be used to trace performance problems of database software services. The step chart and tabular data allow you to analyze query execution details not available in VAS and to trace related problems such as slow queries.

Report contents and usage

The report is composed of a graph and two tables:

- Sub-queries and mean values common to all occurrences of the query.
- A chronological list of all instances of the query whose sub-query components formed the step chart.

Navigation

From the table showing sequence elements, you can drill down to details for individual query instances by clicking the entries in the **Query text** column.

Transaction Log

You can use the **Transaction Log** report to examine time-related statistics for particular transactions.

The **Transaction Log** report can be used to trace performance problems of transactional software services. The step chart and tabular data allow you to analyze transaction details not available in VAS and to trace related problems such as slow transactions

Report contents and usage

The report consists of a graph and two tables:

- Component actions or operations and mean values common to all occurrences of the transaction.
- A chronological list of all instances of the transaction whose operation components formed the step chart.

Operation Log

You can use the **Operation Log** report to examine time-related statistics for particular operations.

The **Operation Log** report can be used to trace performance problems of transactional software services. With the step chart and tabular data, you can analyze operation load details not available in VAS and trace related problems such as slow operations.

Report contents and usage

The report is composed of a graph and a table showing the details of the operation.

Alarms

The report server's problem detection and alarm system features a four-layer architecture with advanced filtering options available at each layer.

Alarms are sent to recipients based on *subscriptions*. Users referred to as alarm *subscribers* can select which alarms they want to receive, apply additional filtering criteria, and select the delivery mechanism.

Alarms can be sent to a specified email address, or can be sent via an SNMP trap. There are also alarms that are generated even if they have no subscribers assigned. Such alarm notifications are recorded in the alarm logs, which store records of all alarms generated.

You can modify the existing alarm definitions (those owned by System) or define new alarms.

NOTE

Because alarm parameter names are automatically parsed, alarm parameters must retain their default names, even if the user interface language is changed.

Overview of the Alarm System

- The alarm mechanism allows you to be proactive rather than reactive when dealing with problems and to remove problems before they start affecting users.
- Defining meaningful alarms requires careful observation and knowledge of the system.
- The alarm system can satisfy various user requirements and operational scenarios.
- The alarm mechanism can be thought of as consisting of three layers: events, alarm states and notifications.
- Alarms can be divided into different types, based on the type of the underlying detector mechanism.
- Alarms can be sent to a specified e-mail address, or they can be sent via SNMP traps.

The benefit of using alarms

The alarm mechanism allows you to be proactive rather than reactive when dealing with problems and to remove problems before they start affecting users.

In the reactive model of dealing with problems, you react to problems reported by your users, for example Web site users.

In such a scenario, the VAS server is monitoring the given Web site and the AMD is measuring page load time for every page, transaction and user, all the time. Then, using the gathered data, the report server displays all details on charts and makes it possible to measure performance and troubleshoot problems.

When problems are reported by users, you look at the reports, find out that, for example, the problem is with HTTP response time from a certain server. You then go and fix the problem, that is reboot or restart the process or take other corrective action. In other words, you react to a problem that *has already affected your users*.

In the *proactive* model, you aim to detect problems *before your users can notice them*. For this you need two things: You need the knowledge of how the problems manifest themselves in your particular environment, and you need to have means of detecting such situations. For example, if in your situation long HTTP response time is the metric that is the best early indicator of developing problems, you could display a chart showing HTTP response time metric and take action if the value of the metric is above certain value.

However, it is even better to automate the process and let the system inform you when the metric exceeds the threshold. This is exactly what the alarm mechanism was designed to do. Ideally, the system could inform a designated operator about the problem, as well as feed data into an alarm management engine. The engine would then perform a corrective action, for example restart the offending server or process.

Thus, the report mechanism allows you to move some of the responsibility and intelligence from a human operator—watching the charts—to the machine.

Defining and modifying alarms

Defining alarms needs to be done with care and you need to make sure that:

- You understand what you are trying to achieve and that you have gathered your requirements.
- You know how problems in the monitored system manifest themselves, though this comes only from experience.
- You are able to translate your intentions into alarm configuration. You must make sure that alarms detect error situations and nothing but error situations. In other words, you must make sure that failure notifications are sent and corrective actions are performed always when needed, but only in those situations.

The above must be done based on careful observation of the system. Therefore, when configuring alarms, first of all you must try to think what the system would be showing if you were troubleshooting a failure in a reactive mode. These could be for example slow pages, HTTP response time, SSL handshake, Stopped Pages, 5xx HTTP errors on login URL or some textual information that need to be captured with Application errors recognition.

Then you need to ask yourself what values for what time duration are still acceptable and what values mean a real problem. Thus, for example 5 minutes of a high server time might not signify a problem yet, but if it stays high for more than 15 minutes it might, particularly if after 30 minutes you also see 5xx HTTP errors. Then you have to react. With this type of information you can start to think about looking for the right alarms to configure.

Note that it is not enough to detect, trigger and send alarm notifications. There need to be a business processes that will ensure this situation will be fixed as soon as possible, for example by restarting a server. In other words, it is not enough to generate many alarms from monitoring tools, if you still react and fix problems only when users call to complain.

Use scenarios for alarms

The alarm system can satisfy various user requirements and operational scenarios, such as:

- Notifying the recipient of both the beginning and the end of the alarm condition: The user is notified when an alarm condition is raised and also when the situation returns to normal.
- Notifying the recipient only if a given condition lasts for a certain period of time, or if a given event is repeated several times. This allows the user to focus on real issues and not on insignificant or intermittent glitches.
- Sending notification to an automated event correlation engine that requires that the error condition notification be maintained throughout the duration of the problem, with an alarm notification repeated at regular intervals.

The Concept of Events, Alarms and Notifications

The alarm mechanism can be thought of as consisting of three layers: events, alarm states and notifications.

Events

Events occur in the monitored traffic, for example a particular metric exceeds its threshold value. Note that an event can be a positive condition such as a given metric exceeding its pre-defined threshold, or an event can be the lack of such a condition occurring: a given metric *not* exceeding its threshold value.

It is important to distinguish events from alarms or notifications. If a given metric exceeds its threshold value, an alarm state might not be triggered immediately. Exactly when an alarm is triggered, is defined in the alarm definition. It often happens that we want to raise an alarm after a threshold has been exceeded a number of times in a given time interval. Similarly, notifications are not sent in direct response to events but in connection with alarm states being raised, remaining on or being lowered.

Alarms states

Alarms can be raised or lowered. A given alarm is either in the *off* (lowered) state or in the *on* (raised) state. The alarm definition specifies when the alarm is raised. This is defined in terms of events. For example, an alarm can be raised:

- As soon as a given event occurs, that means after *one* occurrence of the event.
- After a specified *number* of occurrences of a given event.

An alarm state can then be lowered:

- Immediately after the condition (event) that triggered the alarm has ceased to occur.
- If the triggering event has not reappeared for a specified number of minutes.
- If the triggering event has not reappeared for a specified number of reporting cycles.

It is important to distinguish alarms from events and from notifications: an event can repeat a number of times, but once an alarm is triggered (raised), the alarm condition remains in the *on* state, until the alarm is turned off or expires. Similarly, once an alarm condition is raised, a notification can be sent zero or more times, while the alarm state remains on, and finally a notification can be sent when the alarm is turned off.

Alarm notifications

Once an alarm is raised, a notification can be sent, but does not have to be sent. This depends on the alarm definition. Later on, when the alarm state remains on, repeated notifications can also be sent.

In particular while the alarm remains on, an alarm notification can be repeated:

- in every reporting cycle.
- every specified number of minutes.

It is important to distinguish notifications from events and from alarms: Notifications are not sent in direct response to events, but in response to alarm states being raised, remaining on or being lowered. One alarm can send a number of notifications. Once an alarm is turned on, it remains in the on state, until it is turned off.

Alarm cancellation notifications

An alarm definition can specify that a notification should also be sent when the alarm state is lowered, that is when it reverts to the off state.

NOTE

Threshold values defined for alarm-triggering events are not the same as the performance thresholds set in the server database for the purpose of generating performance reports. The former are used only for the defined alarms and affect only one report: the **Alarm Log Viewer** report, the latter affect the look and values presented on all of the performance reports.

The following graphs illustrate different alarm triggering scenarios.

Figure 32. The simplest case when an alarm is raised is immediately after an event

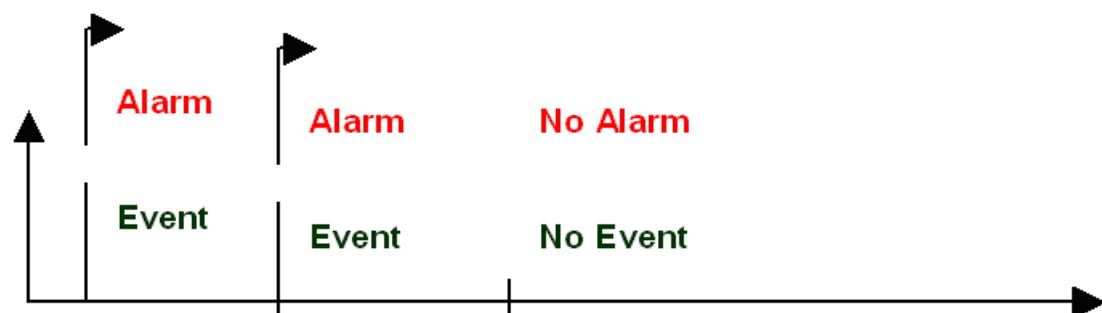


Figure 33. An alarm is activated after a certain number of consecutive occurrences of the triggering event (three). After that, the alarm continues until the events cease to occur

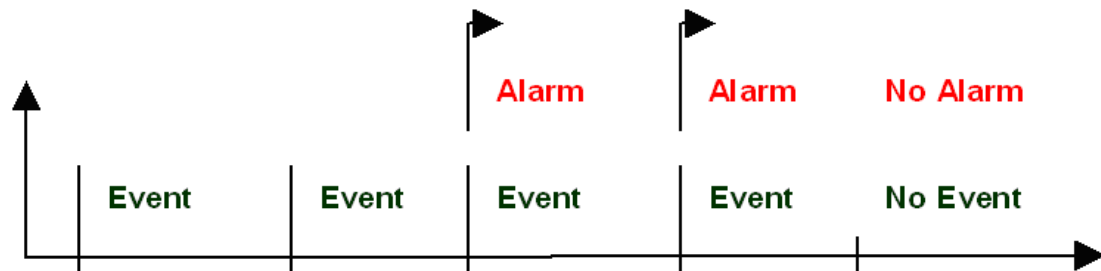


Figure 34. A situation where an alarm is triggered after an event has occurred a number of times in a row (three), and another alarm is triggered when the same event failed to occur for a number of times (two)

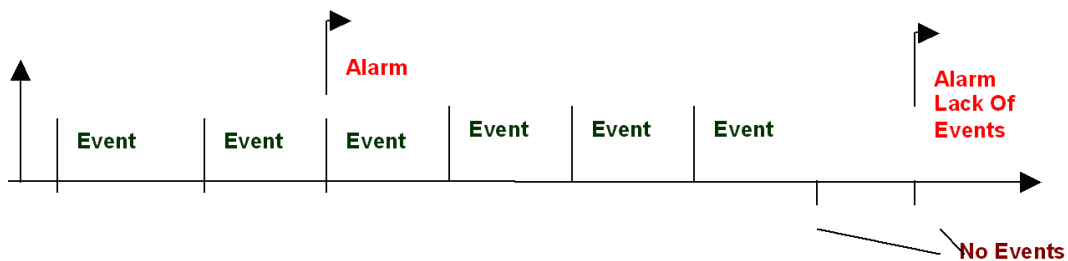


Figure 35. An alarm is triggered after a number of consecutive occurrences of an event (three) and later on, another alarm is triggered after the event failed to occur a number of times in a row (two). In addition, another alarm is triggered a certain time (20 minutes) after the first alarm

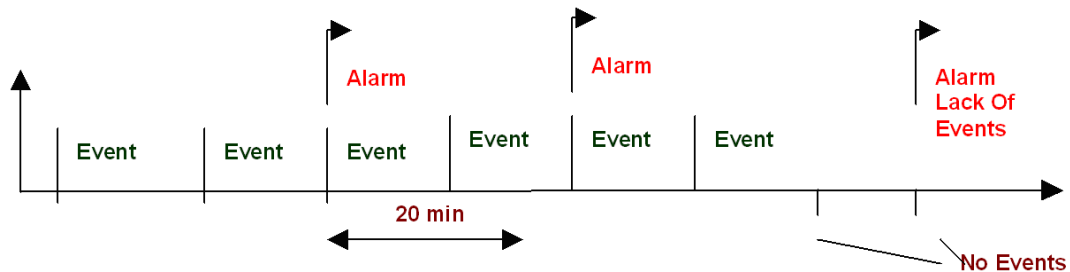


Figure 36. An alarm is triggered after a number of consecutive occurrences of an event (three) and later on, another alarm is triggered after the event failed to occur a number of times in a row (two). In addition, another alarm is triggered after a number of consecutive occurrences of the event since the first alarm was triggered (two). There can be any number of such alarms and there can be a time with no events in-between (see also example below)

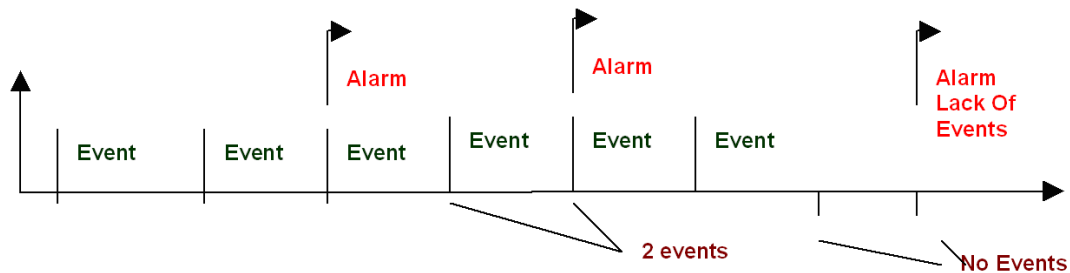
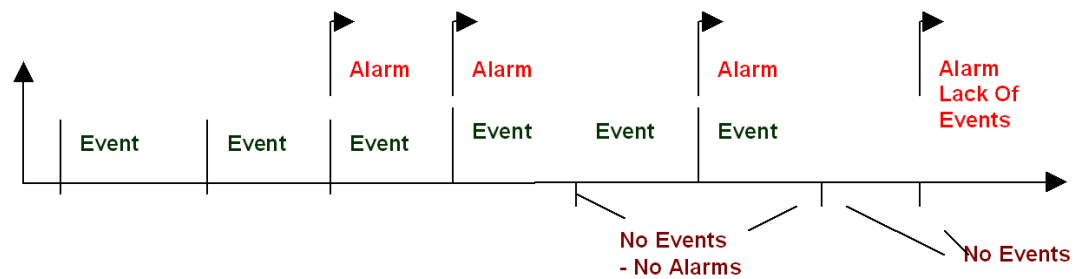


Figure 37. This is the same case as described above except that the alarm that is triggered after a number of occurrences of the event since the first alarm, is triggered after ONE repetition only



Note that there could be a time with no events before the repetitions start.

Also, note that the alarm, which is triggered because of a repetition, is a *separate* alarm for every repetition, and can still be turned on while another instance of the repetition causes another alarm.

Figure 38. An alarm is triggered after a number of consecutive occurrences of an event (three) and later on another alarm is triggered after an event referred to as the "finish event", signaling the end of the original alarm condition. In addition, another alarm is triggered a certain time (20 minutes) after the first alarm

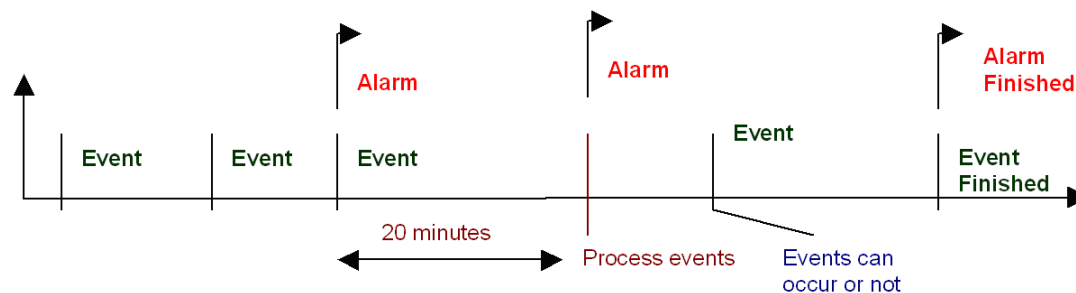
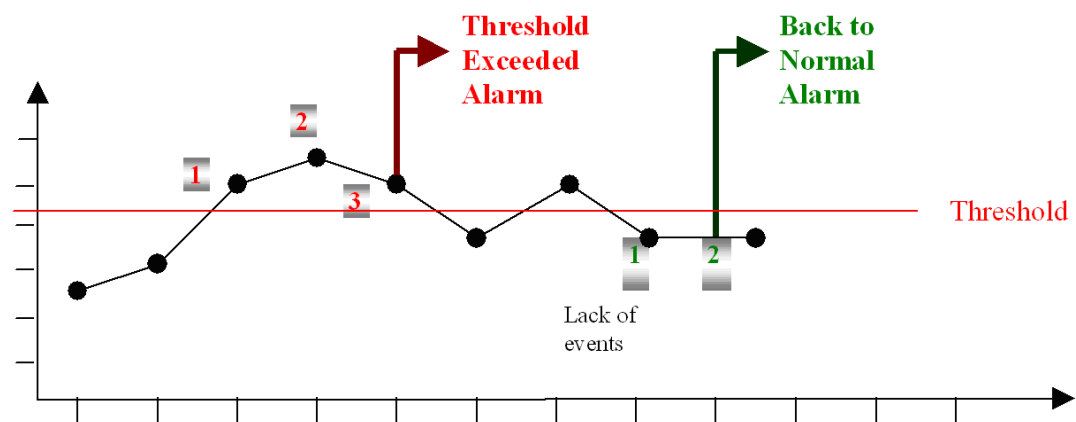


Figure 39. An example graph of a metric related to an alarm triggered after a number of events (three) and another alarm triggered after the event fails to occur a number of times (twice)



In the graph above, events occur when the metric exceeds the threshold value. Lack of events occurs whenever the value of the metric is below the threshold level.

Types of Alarms

Alarms can be divided into different types based on the type of underlying detector mechanism. Alarm detectors are the actual mechanisms responsible for analyzing the monitored traffic and recognizing alarm triggering events. The detector mechanism determines such things as the types and number of parameters that the given alarm takes or can be modified to take, speed of processing, user access to the actual detector code and others. Currently, the following types of alarms are supported:

Alarms based on SQL detectors

These alarms perform queries on the traffic monitoring database, using SQL queries. The benefit of using these alarms is that there are no constraints as to the complexity of the queries and any event that can be expressed as an SQL query can be detected. The drawback of these alarms is that if a modification to the associated SQL code is required, the user must know the structure of the monitoring traffic database, as well must be able to formulate SQL queries. In addition, the SQL queries take not inconsiderable amount of time to execute and performance problems can result.

The system provides a set of pre-defined SQL-based alarms. These can be used as are or they can be modified, copied and saved under new names. It is also possible to define SQL-based alarms from scratch.

Metric alarms based on the METRIC_ALM template (VAS only)

These alarms provide a simple and fast mechanism for performing complex queries on a set of pre-defined metrics. The advantage of using these alarms is easy of use and modification as well as performance. In order to define metric alarms, the user does not need to know the structure of the database and does not need to know how to program in SQL. However, not all conditions can be expressed as metric alarms.

Network alarms based on the METRIC_ALM_P2P template (VAS only)

These alarms are similar in design and function to the metric alarms above, though they view the monitor traffic as it is done on the **Network View** report.

Link alarms based on the LINK_ALM template (VAS only)

Link alarms are based on the LINK_ALM template. They are fast executing alarms that are designed to monitor link utilization, as presented on the **Link View** report.

Other alarms

Apart from the above types of alarms, there are a few other alarms designed for very specific purposes. These alarms can be modified in only a limited way and they do not allow user access to the detector code. Consequently, the number and meaning of detector parameters for these alarms cannot be changed.

NOTE

Not every possible alarm condition can be expressed as a metric alarm, and therefore the conventional SQL alarms are provided. However, it is recommended that metric alarms be used whenever possible, because of their speed of execution and ease of modification. For Network View and Link View monitoring, the corresponding alarm types should be used.

Means of Alarm Delivery

Alarms can be sent to a specified e-mail address, or can be sent via SNMP traps. There are also alarms that are generated even if they have no subscribers assigned to them. To view such alarm notifications, user need to consult alarm logs: all alarm notifications, those e-mailed and those with no subscribers, are recorded in alarm logs.

Alarms are sent to recipients based on *subscriptions*. Users referred to as alarm *subscribers* can select which alarms they want to receive, can apply additional filtering criteria and select the delivery mechanism. For more information, see [Sending Alarm Notifications by E-mail](#) [p. 76].

When e-mail is the selected delivery mechanism, all alarms that have occurred within a single monitoring interval, can be sent in one e-mail message. To select this option, select the **Aggregation** check box on the **Alarm Notifications** screen. This option is selected by default.

When traps are the selected delivery medium, there is a separate trap associated with each alarm notification. For each trap, there is an associated trap definition, identified by an OID, in the MIB in the `alarms.mib` file. This MIB can be imported on the trap recipient, in order to correctly interpret the meaning of the alarm, and automate the corrective actions if such are defined. Refer to your network management platform manual for information on how to install third-party MIBs.

The Process of Defining an Alarm

Defining an alarm is a process that begins with identifying a need for an alarm, and then goes on to defining alarm settings and arranging for the alarm message to be sent to the correct audience. It is useful to follow a top-level procedure to make sure that all the required steps have been followed.

1. Identifying a business need for an alarm

For a detailed discussions of what an alarm mechanism could be used for, see [Overview of the Alarm System](#) [p. 67]. For a detailed example of how an alarm need can be identified and converted to an actual alarm, see *Alarm Usage Example in a Web-based Environment* in the *ClientVantage Agentless Monitoring – System Administration Guide* and *Alarm Usage Example in an Enterprise Environment* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

2. Choosing and enabling an alarm definition

You will need to review the list of existing alarms and select or create one that best suits your need. You can then modify this alarm definition or create a new definition based on it.

To make a choice appropriate for your need, consult the descriptions of the pre-defined alarms as given in *Alarm Definitions* in the *ClientVantage Agentless Monitoring – System Administration Guide*, and for metric alarms see the list of available metrics in *Metrics Available for Metric Alarm Definitions* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

Once you have made your choice, you will need to access and enable the correct definition. For information on how to manage, enable or disable individual alarms, see *Managing Alarms* in the *ClientVantage Agentless Monitoring – System Administration Guide* and *Modifying Alarm Definitions and Creating New Alarm Definitions* in the *ClientVantage Agentless Monitoring – System Administration Guide*. Special considerations for metric alarms are described in *Special Considerations for Defining Metric Alarms (VAS Only)* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

3. Modifying the required detector parameters.

You will most likely need to review the detector parameters and modify some of the threshold values as described in *Modifying Parameter Values of Alarm Detectors* in the *ClientVantage Agentless Monitoring – System Administration Guide* and shown in the examples *Alarm Usage Example in a Web-based Environment* in the *ClientVantage Agentless Monitoring – System Administration Guide* and *Alarm Usage Example in an Enterprise Environment* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

4. Defining propagation characteristics

It is important to send alarm notifications in the correct circumstances and with correct frequency. For more information, see *Defining Propagation Characteristics* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

NOTE

If you wish for the alarm to appear in the alarm log even if there is no defined alarm recipient, you will need to specify that the alarm should be *unmatched with a recipient*.

5. *Optional*: Editing alarm fields

Alarm fields are numeric or string output values returned by alarm detectors, in alarm notifications. These output fields can be referred to by user-defined names, which are used on alarm configuration and notification screens as well as in MIBs—if alarm notifications are sent using traps.

6. *Optional*: Editing the alarm message template

If you need to customize the alarm message, follow the steps described in *Editing Alarm Message Template* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

7. *Optional*: Specifying alarm annotations

Annotations specify circumstances (related to the alarm propagation characteristics) under which the given alarm notification was sent. You can specify them as described in *Specifying Annotations* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

8. *Optional*: Specifying alarm filters

Additional conditions can be set on alarm output fields, so that only alarms satisfying those conditions are raised. To specify those, follow the steps described in *Specifying Alarm Filters* in the *ClientVantage Agentless Monitoring – System Administration Guide*

9. Optional: Providing an alarm description

You should provide an alarm description when defining an alarm to help you to keep track of alarm functionality. For more information, see *Providing an Alarm Description* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

10. Optional: Specifying the Mail Server

To send alarm notifications via e-mail, a user with report server administrator privileges must configure the server to use an existing SMTP server. For more information, see *Specifying the Mail Server* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

11. Optional: Defining alarm e-mail recipients

If alarm notifications are to be delivered by e-mail, you need to define report server users with those e-mail addresses. For information on how to define new users or edit existing user definitions, see *Adding New Report Server User* in the *ClientVantage Agentless Monitoring – System Administration Guide* and *Modifying User Account Details* in the *ClientVantage Agentless Monitoring – System Administration Guide*

12. Optional: Arranging for alarm notifications to be e-mailed to selected users

For more information, see [Sending Alarm Notifications by E-mail](#) [p. 76].

13. Optional: Arranging for alarm notification to be sent to Compuware Open Server (COS)

For more information, see *Alarm Delivery to Compuware Open Server (COS)* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

14. Optional: Specifying trap clients

To receive traps generated by the report server, the administrator has to specify trap recipients, also referred to as trap clients. For more information, see *Specifying Trap Clients* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

15. Optional: Configuring traps for sending alarms

For more information, see *Configuring Traps for Sending Alarms* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

Sending Alarm Notifications by E-mail

Alarm notifications can be delivered by e-mail to the e-mail addresses specified for report server users.

To send alarm notifications via e-mail, a user with administrator privileges must configure the report server to use an existing SMTP server. For more information, see *Specifying the Mail Server* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

After you have configured the mail server, you need to specify which alarms are to be sent to which users. To specify alarms notifications to be e-mailed to the *currently logged* user, select **Settings** → **Alarms** → **Alarm Notifications** from the menu bar to open the **Alarm Notifications** screen. To specify alarm notifications to be sent to *other* users, you should enter the same **Alarm**

Notifications screens, but for those particular users. To do this, you should first invoke the **User Administration** screen, and then select the required users and click the **Alarms** button. For more information, see *User Administration and Security* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

Figure 40. An example of the **Alarm Notifications** screen

Alarms for User : superuser

ID	Description	E-mail	Aggr
MQ_SLOW_SE	Slow Message Queue operations d	OFF	
NEW_APP	New software service detected	OFF	
NEW_SERVER	New server detected	OFF	
NEW_SERVIC	New service detected	OFF	
NEW_USER	New user detected	OFF	
NEW_WORKS	New workstation detected	OFF	
OF_SLOW_FO	Slow Oracle Forms submissions d	OFF	
OF_SLOW_SE	Slow Oracle Forms submissions d	OFF	
P2P_VOIP_MO	High P2P bad MOS calls over VoIP	OFF	
P2P_VOIP_MO	Low P2P MOS over VoIP software s	OFF	
PAGE_LOAD	Long page load time for URL	OFF	
RBAND	Realized bandwidth too low for site	ON	ON
SERV_PERF	High server time for service	OFF	
SRV_ERR_GR	HTTP server errors unacceptable fc	OFF	
SSL_APPL_IN	SSL connection setup time too long	OFF	
SUSP_CLI_TR	Multiple IP addresses used by user	OFF	
SUSP_URL_T	Abnormal URL traffic for software s	OFF	
SVR_TIME_4	High server time for URL pattern oc	OFF	
SYS_STATUS	Server status	OFF	
TFC_LVL	Traffic level out of bounds for server	OFF	
TFC_SUSP	Suspicious traffic characteristics or	OFF	
URL_RESP_E	Server time and number of slow pa	OFF	
USER_AVAIL	Service availability problem for user	OFF	
VPN_DROP_O	VPN gateway drop off	OFF	

Realized bandwidth too low for site

☒ Send alarm by e-mail

☒ Aggregation

☒ Deliver also 'Alarm Finished' events

Site

Normal realized bandwidth [bps]

Current realized bandwidth [bps]

Number of users

☐ Deliver all generated alarms ☒ Show disabled

To activate an alarm, select the alarm in the list and select the **Send alarm by e-mail** check-box on the right hand-side. Note that this option will not work unless the administrator adds and configures the recipient's e-mail address.

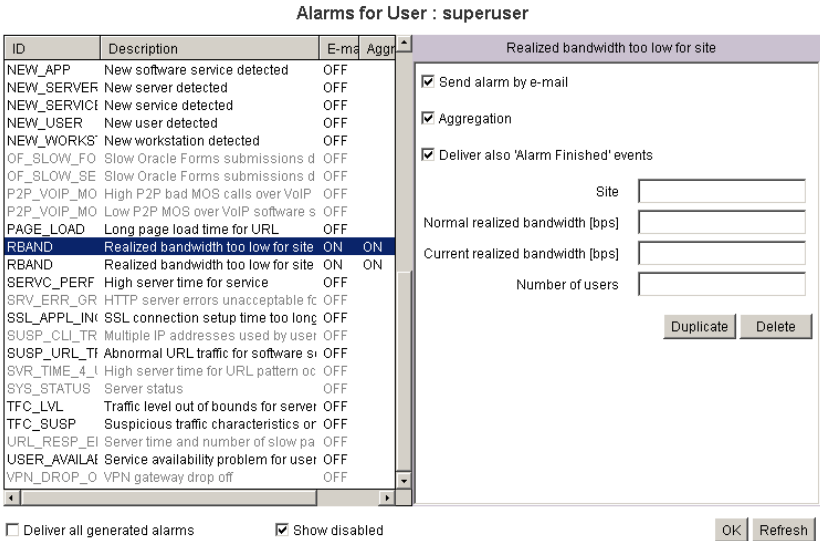
For some alarm types, you can also select the **Deliver also "Alarm Finished" events** check-box to notify by e-mail when the condition that triggered the alarm returns to normal.

The **Aggregation** check box (selected by default) makes sending alarm notification more efficient by combining multiple alarm messages into one e-mail message, when the alarm situation occurs more than once in a monitoring interval.

The remaining input fields enable you to refine the conditions that cause a notification to be sent. The fields accept explicit values or expressions representing sets or ranges of values. Note that these conditions affect only alarm notifications sent by e-mail. The **Alarm Log Viewer** report is unaffected by these filters (all alarms are recorded when alarm situations occur).

Each alarm can be duplicated by clicking the **Duplicate** button. This will copy the selected alarm item.

Figure 41. An example of a duplicated alarm screen



Each copy of the duplicated alarm assignment can have different filtering criteria but all are based on the same alarm definition.

To delete an alarm copy, select it and click **Delete**.

To undo changes, click the **Refresh** button. This will cause all the alarm settings to revert to the state stored on the server.

Click **OK** to confirm all alarm assignments.

The Contents of Alarm Messages

Alarm notification messages are sent by e-mail with the message subject Vantage Analysis Server Alarm Engine and contents such as the following:

```
Timestamp: date and time

Realized bandwidth too low:
Affected location: Default
Normally: 29500 bps
Currently: 17407 bps
Affected users: 15

Realized bandwidth too low:
Affected location: MY Location
Normally: 141208 bps
Currently: 84205 bps
Affected users: 9

Realized bandwidth too low:
Affected location: Sienna AS
Normally: 75412 bps
Currently: 45104 bps
Affected users: 3
```

In the case of notifications sent by traps, there are three separate messages, corresponding to the above three text segments.

APPENDIX A

Protocols Supported by AWDS

Table 2. Supported protocols in this release

Analyzer	Protocol	Version	Limitations / example application
HTTP	HTTP	1.0, 1.1 (RFC 2616)	Advanced analysis for GET/POST methods. For all other methods, including WebDAV, every hit is reported separately. No pipelining.
Oracle Applications over HTTP ¹	HTTP	1.1, 1.0 (RFC 2616)	Oracle E-Business Suite 11i Oracle E-Business Suite 12
Oracle Applications over HTTPS ¹	HTTPS	HTTP 1.1 encapsulated in SSL, SSL 3.0, TLS1.0 (RFC 2246)	
Oracle Forms (over HTTP) Oracle Forms (over TCP) Oracle Forms (over SSL) Oracle Forms (over HTTPS)	Oracle Forms	6i, 10.1	Oracle Forms 6i Oracle Application Server 9i, 10i, 10g R2 Transactions not supported.

¹ Monitoring applications using HTTP protocol may register excessive traffic. For more information, see *Supported Packaged Applications* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

Analyzer	Protocol	Version	Limitations / example application
SAP GUI	SAP GUI protocol (DIAG)	6.40, 7.10	No errors detection. SAP GUI for Java 7.10rev8, Windows SAP GUI v.7.10, SAP GUI Console
Siebel over HTTP ² Siebel over HTTPS ²	HTTP	1.1, 1.0 (RFC 2616)	Siebel CRM 7.8.2.0
Siebel over HTTPS ²	HTTPS	HTTP 1.1 encapsulated in SSL, SSL 3.0, TLS1.0 (RFC 2246)	
SSL (with decryption)	HTTPS	HTTP 1.1 encapsulated in SSL, SSL 3.0, TLS1.0 (RFC 2246)	56-bit DES is not supported. Only RSA Key Exchange Algorithm supported. GET/POST methods only; no pipelining.

² A special parameter configuration is recommended for analyzing Siebel applications. For more information, see *Global Settings for Recognition and Parsing of URLs* in the *ClientVantage Agentless Monitoring – System Administration Guide*.

Dimensions and Metrics

Page Analysis data view

Page Analysis data view dimensions

Agent

The name of the synthetic agent that loaded the HTTP page(s), for example, Keynote, Gomez, or Mercury. The name of the agent is determined from the User-agent field of the HTTP request and/or from agent user names or IP address configured on the server.

AMD UUID

UUID (Universal Unique Identifier) of the AMD that produced the data.

Analyzer group

An analyzer group name. For example: Web, XML or Oracle Forms.

Client area

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names, CIDR blocks or subnets. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client AS

The name of the client's BGP Autonomous System.

Client ASN

Autonomous System Number (ASN) of the client site

Client city

Geographical data about the client site.

Client country

Geographical data about the client site.

Client geographical region

Geographical data about the client site.

Client group

The client's group, as manually defined in Vantage Analysis Server.

Client IP address

The IP address of the client.

Client IP address (internal)

Client IP extracted from the HTTP header on the AMD side.

Client port

The TCP port number on a client machine that hosts a software service.

Client region

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client site

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions, clients' BGP Autonomous System names, CIDR blocks or subnets. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client site description

Optional description of the client site.

Client site ID

In cases when sites are ASes, Client Site ID contains the AS number, which is also given in Client ASN. For manual sites, Client Site ID is identical to Client site, and contains the site name as defined in your site configuration. Sites based on CIDR blocks or subnets are identified by IP addresses.

Client WINS name

The client's computer name resolved by a WINS server

Correlation ID

A unique numerical identifier of an XML/SOAP message, assigned by the AMD, used for linking messages into one transaction in the Transactions, Page Analysis and Page Elements data views.

Direction

The role that the page played in the transaction:

- 0 - Request (HTTP, XML traffic)
- 1 - Response (XML traffic)

HTTP application response messages

The messages of categories 1-5, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category1 (default name)

The messages of category 1, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category2 (default name)

The messages of category 2, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category3 (default name)

The messages of category 3, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category4 (default name)

The messages of category 4, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category5 (default name)

The messages of category 5, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP response status

Response status as read from the server response header. DMI also displays status name as a result of the query. Value -1 means 'No response header', other values as from the definition of the HTTP protocol.

Operation ID

A unique numerical identifier of a page load assigned by the AMD, used for linking page hits into one page on the Page Elements data view. Note that when selecting the Operation ID as a dimension, you cannot specify more than one value in the Condition/Negate field.

Page begin time

In addition to the measurement (AMD batch of data) time, each page is separately stamped with its actual time of occurrence, with 1-millisecond granularity. This dimension can be used to list pages one by one, in the order of occurrence, to reflect exactly the client or server activity, page by page.

Page elements index

The time stamp of the data presented on the report.

Page name

Alias for the page URL set on AMD's side or page URL itself if alias is undefined.

Page status

It can be equal to Abort or No Abort. No Abort means that the page has been downloaded successfully. Abort indicates that the following error(s) might have occurred:

- Client Abort
- Break (TCP Reset)
- Error (Timeout)
- Dead (The hit idling for too long)

Page URL

The URL of the page. This is ascertained by the AMD, based on referrer, timing relations between hits and per-transaction monitoring configured on the AMD. This dimension can assume values of a particular URL, if this URL is monitored. Note: The visibility of this dimension on reports depends on whether another dimension, related to servers - e.g. server IP or server DNS - has been used when formulating the query definition.

Parent reporting group

An application that acts as a "container" for a specific transaction.

Protocol

IP protocol name.

Reporting group

Reporting group is a universal container that can accommodate software services, servers, URLs or any combination of these. Reporting groups can contain software services of every type. Advanced Web Diagnostics Server can import reporting group configuration from Vantage Analysis Server.

Request method

HTTP request type: GET or POST.

Root cause

Root cause of slow page load. Page Analysis distinguishes the following types of the root cause:

- Desktop
- Network
- Page design
- Data center
- Other

Root cause details

The detailed information on the detected root cause.

Server area

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and optionally on clients' Autonomous System names, CIDR blocks or subnets. Sites are the smallest logical structures of clients and servers. Areas are composed of sites. Regions are composed of areas.

Server AS

The name of the Autonomous System Name of the server site.

Server ASN

Autonomous System Number (ASN) of the server site.

Server city

Geographical data about the server site.

Server country

Geographical data about the server site.

Server DNS name

The name of the server resolved by a DNS server.

Server geographical region

Geographical data about the server site.

Server host name

The name of the server, as read from the Host field of the packet header.

Server IP address

The IP address of the server.

Server port

The TCP port number on a server that hosts a software service.

Server region

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Server site

Sites, areas, and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' Autonomous System names. Sites are the smallest logical structures that comprise of clients and servers. Areas are composed of sites, while regions are composed of areas.

Server site description

Optional description of the server site.

Server site ID

In cases when sites are ASes, Server site ID contains the AS number, which is also given in Server ASN. For manual sites, Server site ID is identical to Server site, and contains the site name as defined in your site configuration. Sites based on CIDR blocks or subnets are identified by IP addresses.

Software service

Software service name, where by a software service we understand a service implemented by a specific piece of software, offered on a TCP or UDP port of one or more servers and identified by a particular port number.

Time

The time stamp of the data presented on the report.

Traffic type

The type of client traffic: real or synthetic, that is, generated by a synthetic agent.

Transaction ID

A unique identifier of a transaction to which this page belongs. A transaction is a sequence of page loads used for accomplishing a particular action, for example on-line travel or appointment booking, on-line shopping etc.

User name

Client's name determined from HTTP cookie (requires configuration on AMD), HTTP authentication header or static mapping.

Page Analysis data view metrics

50pc network time

The 50th percentile of the network time.

50pc page load time

The 50th percentile of the page load time.

50pc server time

The 50th percentile of the server time.

80pc network time

The 80th percentile of the network time.

80pc page load time

The 80th percentile of the page load time.

80pc server time

The 80th percentile of the server time.

85pc network time

The 85th percentile of the network time.

85pc page load time

The 85th percentile of the page load time.

85pc server time

The 85th percentile of the server time.

90pc network time

The 90th percentile of the network time.

90pc page load time

The 90th percentile of the page load time.

90pc server time

The 90th percentile of the server time.

95pc network time

The 95th percentile of the network time.

95pc page load time

The 95th percentile of the page load time.

95pc server time

The 95th percentile of the server time.

99pc network time

The 99th percentile of the network time.

99pc page load time

The 99th percentile of the page load time.

99pc server time

The 99th percentile of the server time.

Aborts

The number of pages aborted by the client.

For example, for HTTP/HTTPS, it is the number of page loads manually stopped by the user by either clicking on the Stop or Refresh buttons or selecting another URL. Note that, in the case of HTTP, this number includes Short aborts and Long aborts.

Affected users

The number of users who experienced slow page loads.

AMD storage delay

The AMD delay in reporting the page load.

Apdex

This metric is based on the APDEX performance index (Application Performance Index) as defined at www.apdex.org. It converts the available information about application response times into one number on a uniform scale of 0-to-1, where 0 means that no users were satisfied, and 1 means that all users were satisfied with the performance of the application. The metric, in the context of Page Analysis, applies to page load times only. The performance index attempts to reflect the actual perceived performance, and aims to avoid the problems related to assessing the user experience by using raw, absolute values of response times. It also attempts to overcome the problems of loss of detailed information that occurs if response times are averaged.

Client bandwidth usage

The number of client bits per second.

Client bytes

The number of bytes sent by the client(s). Note that this includes headers.

Client loss rate

The percentage of total packets sent by a client that were lost (due to network congestion, router low queue capacity or other reasons) and needed to be retransmitted.

Client packets

The number of packets sent by the client.

Client RTT

Client RTT is the time it takes for a SYN packet (sent by a server) to travel from the AMD to the user and back again.

Connection setup time

The time required to set up an HTTP connection between the client and the server, that is the time difference between the last packet of the HTTP request and the SYN packet that was sent by the client, at the start of the TCP session.

Custom metric (1) (avg) (default name)

The average value of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (1)(cnt) (default name)

The number of occurrences of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (1) (sum) (default name)

The sum of all values of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (2) (avg) (default name)

The average value of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (2)(cnt) (default name)

The number of occurrences of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (2) (sum) (default name)

The sum of all values of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (3) (avg) (default name)

The average value of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (3)(cnt) (default name)

The number of occurrences of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (3) (sum) (default name)

The sum of all values of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (4) (avg) (default name)

The average value of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (4)(cnt) (default name)

The number of occurrences of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (4) (sum) (default name)

The sum of all values of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (5)(cnt) (default name)

The number of occurrences of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Custom metric (5) (sum) (default name)

The sum of all values of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Custom metric (51) (avg) (default name)

The average value of user-defined metrics in category 5 observed in the HTTP or XML traffic.

End-to-end RTT

The time it takes for a SYN packet to travel from the client to a monitored server and back again.

Hits

The number of subcomponents of error-free operations or transactions. Note that this metric is recorded at the time when the monitored transactions are closed. In case of HTTP, it is when the whole page has been loaded. Compare "Hits (started)". For example, when a user issues an HTTP Get, this is reported immediately as a "Hit (started)". If this does not result in an HTTP error, it is recorded as a "Hit".

Hits per page

The number of HTTP hits per page.

HTTP application responses

The number of HTTP application responses of types 1-5, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 1 (default name)

The number of HTTP application responses of type 1, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 2 (default name)

The number of HTTP application responses of type 2, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 3 (default name)

The number of HTTP application responses of type 3, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 4 (default name)

The number of HTTP application responses of type 4, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 5 (default name)

The number of HTTP application responses of type 5, observed for the HTTP- or HTTPS-based software service.

HTTP client errors (4xx)

The sum of all HTTP client errors (4xx). This includes 4 categories of errors (4xx), by default HTTP Unauthorized (401, 407) errors, HTTP Not Found (404) errors, custom client (4xx) errors and Other HTTP (4xx) errors. The contents of the first 3 categories can be configured by users.

However, there are two types of the 4XX errors that are of particular importance: errors 401 related to server-level authentication, and errors 404 indicating requests for non-existent content. These two error types are reported separately, by specific metrics.

- 401 Unauthorized - Server reports this error when user's credentials supplied with request do not satisfy page access restrictions. The HTTP server layer, not the

application layer, reports 401 errors. The AMD will report on "Unauthorized" errors only if server-level authentication has been configured. This is common practice for sites that are comfortable with very basic user access policies. Most commercial-grade applications do not rely on server-level authentication (e.g. most of online banking applications or online shopping), but rather authenticate users on the application layer. In such a case, even if authentication fails, the server will typically send 200 OK responses and authentication error information will be explained in page content. So this kind of error is not very common in commercial sites.

- **404 Not Found** - Server reports "Not Found" errors when it cannot fulfill client request for a resource. Usually it happens due to malformed URL, which directs to a non-existing page or image. Such a URL request may result from a user, who misspelled the URL, trying to access a URL that the user stored in his "Favorites" folder a long time ago, or some other mistake. Malformed URLs may also exist in invalid or incorrectly designed Web pages so the error will be reported by browsers trying to load such a page. Significant and constant number of these errors usually indicates that some pages on the server have design-related or link validation issues. In some cases, 404 errors result from the server overload. It is good practice to check if the percentage of errors is load-related.

HTTP client errors (sumCliErr_B1)

The number of custom defined HTTP client errors (4xx), category 3.

HTTP client errors (sumNotfoundErrPage)

The number of observed HTTP 404 Not Found errors.

HTTP client errors (sumUnautErrPage)

The number of observed HTTP 401 Unauthorized or HTTP 407 Proxy Authentication Required errors. For more information on HTTP 401 errors, see definition of the metric returning the number of the error occurrences.

HTTP errors

The number of HTTP hits with errors.

HTTP errors per page

The number of page elements that returned HTTP errors in the 4xx and 5xx ranges. This metric is an average number of errors per page.

HTTP other client errors (4xx)

The number of HTTP other client errors (4xx).

There are four categories of HTTP client errors (4xx), of which three can be configured by users. By default, the first category includes HTTP Unauthorized (401, 407) errors, the second category - HTTP Not Found (404) errors. The third category contains no default error types assigned, and can be configured by a user. Finally, a group of HTTP Other (4xx) errors contains all errors that do not fall into any other client errors category. The number is calculated based on the formula: [HTTP errors 4xx] - [HTTP Not Found errors 404] - [HTTP Not Authorized (401+ 407)] - [HTTP errors configured by user]

HTTP other server errors (5xx)

The number of HTTP server errors (5xx) that do not fall into categories 1 or 2 of custom HTTP server errors (5xx).

HTTP request time

The time elapsed between the first and the last packet of the HTTP or database request. Note that this time does not include the connection setup time.

HTTP server errors (5xx)

The number of observed HTTP server errors (5xx).

The response status codes 5xx indicate cases, in which the Web server is aware that there was a server error or it is incapable of performing the request. Such error presence usually means that the Web server does not function as intended. The following 5xx errors are defined by the HTTP protocol standards:

- 500 Internal Server Error - The server encountered an unexpected condition, which prevented it from fulfilling the request.
- 501 Not Implemented - The server does not support the functionality required to fulfill the request.
- 502 Bad Gateway - The server received an invalid response from a back-end application server.
- 503 Service Unavailable - The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.
- 504 Gateway Timeout - The server did not receive response from a back-end application server.
- 505 HTTP Version Not Supported - The server does not support the HTTP protocol version that was used in the request message.

HTTP server errors (sumSrvErr_B1)

The number of HTTP server errors (5xx), category 1. By default, there is no specific error category assigned to this group.

HTTP server errors (sumSrvErr_B2)

The number of HTTP server errors (5xx), category 2. By default, there is no specific error category assigned to this group.

HTTP server time

The time it took for the server to provide an initial response to the page request or database request. Servers may respond with initial information quickly, before all requested information is ready. HTTP Server Time is equal to or smaller than Server Time.

Idle time

Time during the page load when there is no network or server activity related to the page. It is assumed that Idle time is caused by the user's browser not sending requests because user's PC is busy.

Loss rate

The percentage of retransmitted packets. This includes both the server packets and the client packets.

Max client loss rate

The maximum of the client loss rate.

Max client RTT

The maximum of the client RTT.

Max end-to-end RTT

The maximum of the end-to-end RTT.

Max loss rate

The maximum of the loss rate.

Max network time

The maximum of the network time.

Max number of request cookies

Maximum number of the key-value pairs in the Cookie field of the HTTP request header.

Max number of response cookies

Maximum number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Max page load time

The maximum of the page load time.

Max redirect time

Maximum time of the introductory HTTP redirect.

Max request body size

Maximum of the size of the HTTP request body (sum for page hits).

Max request cookie size

Maximum length of the Cookie field content, including separators.

Max request header size

Maximum length of the HTTP request header (sum for page hits).

Max request time

The maximum of the request time.

Max response cookie size

Maximum length of the Set-Cookie field content, including separators (sum for page hits).

Max response download time

The maximum of the response download time.

Max response header size

Maximum length of HTTP response header (sum for page hits).

Max server loss rate

The maximum of the server loss rate.

Max server RTT

The maximum of the server RTT.

Max server time

The maximum server time.

Min client loss rate

The minimum of the client loss rate.

Min client RTT

The minimum of the client RTT.

Min end-to-end RTT

The minimum of the end-to-end RTT.

Min loss rate

The minimum of the loss rate.

Min network time

The minimum of the network time.

Min number of request cookies

Minimum number of the key-value pairs in the Cookie field of the HTTP request header.

Min number of response cookies

Minimum number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Min page load time

The minimum of the page load time.

Min redirect time

Minimum time of the introductory HTTP redirect.

Min request body size

Minimum of the size of the HTTP request body (sum for page hits).

Min request cookie size

Minimum length of the Cookie field content, including separators (sum for page hits).

Min request header size

Minimum length of the HTTP request header (sum for page hits).

Min request time

The minimum of the request time.

Min response cookie size

Minimum length of the Set-Cookie field content, including separators (sum for page hits).

Min response download time

The minimum of the response download time.

Min response header size

Minimum length of HTTP response header (sum for page hits).

Min server loss rate

Minimum server loss rate.

Min server RTT

Minimum server RTT.

Min server time

The minimum of the server time.

Network time

The time the Internet (between the user and the server) takes to deliver requests to the server and to deliver page information back to the user. In other words, Network time is the portion of transaction time (or transaction time for non-HTTP protocols) that is due to the delivery time on the network/Internet.

Number of request cookies

The number of key-value pairs in the Cookie field of the HTTP request header.

Number of response cookies

The number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Orphaned redirects

The number of HTTP redirects for which a matching request to the target URL was not detected before the timeout time.

Page interval index

The sum of the page load time and AMD delay in reporting the page load.

Page load time

The page load time is a compound metric calculated based on the analysis of the page being measured. The page load process consists of two stages: a) establishing the HTTP session with the Web server, b) actual page load from the Web server.

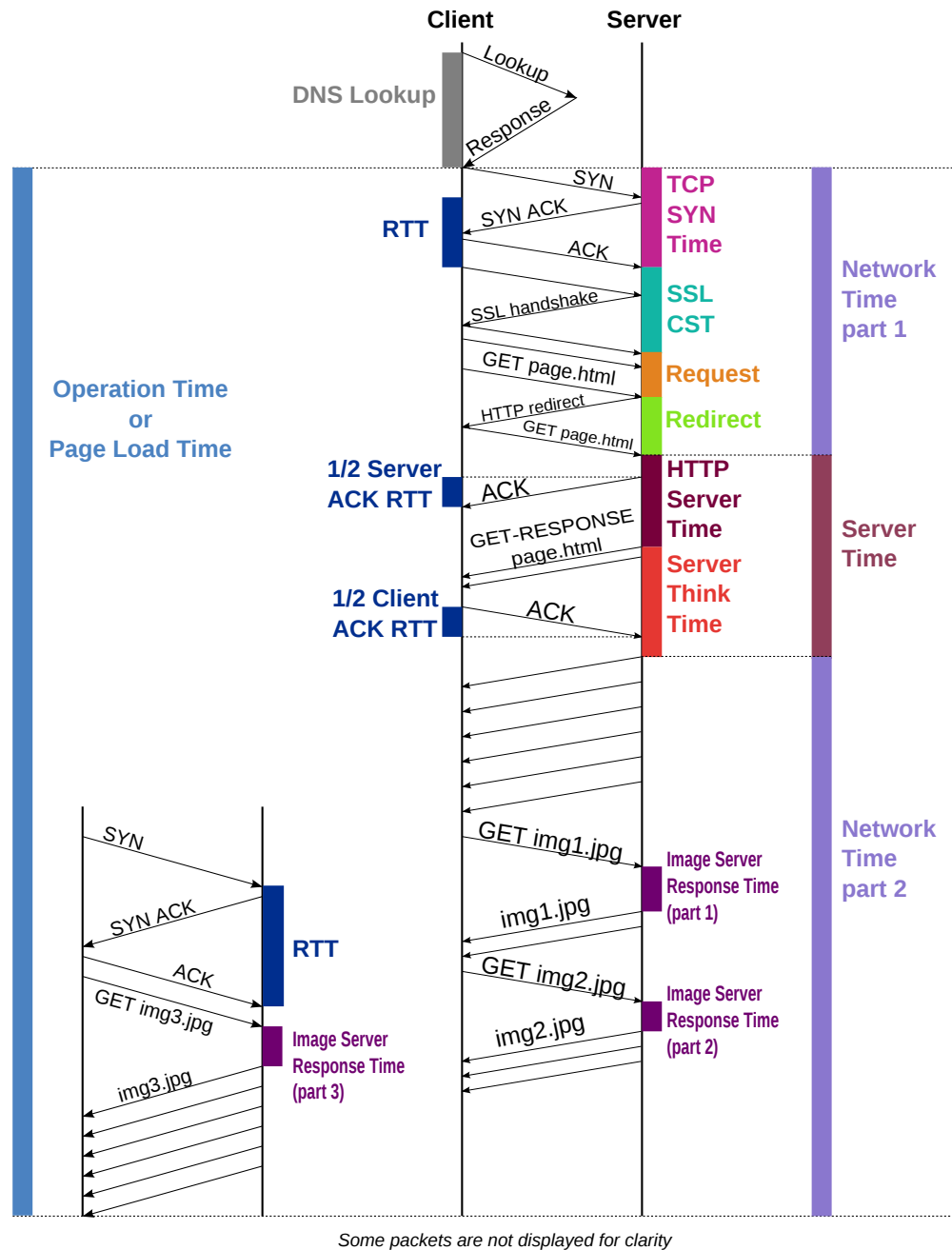
A typical Web page is composed of multiple objects, each of which is retrieved from the server through a single HTTP-level operation or a 'hit'. The duration of each hit is of lesser importance than the complete page load time. To determine page load time, it is necessary to watch all individual HTTP operations (hits) that belong to the page.

- There is an additional factor, called 'Redirect'. When the server gets the request from the browser, it may decide to 'redirect' that browser to another server or another URL. For example, the user may type in a URL `http://www.company.com/page.html` and the server would instruct the browser to fetch `http://www.company.com/pageX.jsp` instead. Redirect operation is optional; it may or may not happen and may be more complex than described here. For example, the browser may be redirected to another server and another DNS Lookup and TCP Connection Setup may be required during that process. The time required to complete that whole process is represented by the Redirect bar on the page load diagram.
- Base page request. When the TCP and optional SSL connections are in place, and after the optional Redirect operation has completed, the browser will request the 'base page', which is an HTML document. Before the server responds, the HTTP stack has to process the request. That time is marked as 'HTTP Server Time' on the diagram.
- Preparation of the base page response In many cases, the page content will be dynamically created based on history of user visits and his preferences, or some other criteria. This requires the HTTP server to involve some kind of a higher-level application, which would produce such customized content. This may be accomplished through a CGI script, or may require a separate application server or potentially a database server to be involved. Dynamic nature of some Web pages is determined by the AMD by looking at the server behavior when it sends the response to the

client. If the AMD detects that server is delaying response delivery despite the fact that client acknowledged the reception of all packets, the AMD marks this event as 'Server Think Time'.

- **Response download** Now the content is being transferred to the browser. After having parsed the document content (or during this process), the browser will open additional TCP connections to the target server. For example, the base page 'page.html' on the page load diagram contains three objects, img1.jpg, img2.jpg, and img3.jpg. To load these objects, the browser opened one additional TCP connection. The number of these additional connections depends on browser type and number of embedded objects. The efficiency of the HTTP protocol depends on the ability to load page elements on concurrent, parallel connections. In such a case, when the site is busy servicing a request for one object, the transfer of another object may utilize the available bandwidth between the browser and the Web server. This transfer time is named 'Network Time Part 1'.
- **Preparation and download of the content elements** In some cases, the elements embedded in the base page may be produced dynamically or may be delivered from dedicated cache servers –in many cases, placed behind content switches. Thus, it makes sense to measure the time spent in the data center to produce such content. The total time represented by 'Image Server Response Time'.

After the page has been successfully loaded, the AMD stores the complete Page Load Time and associated metrics for further aggregation and analysis.



Page load time breakdown

The breakdown of total page load time, as perceived by the user, into redirect time, server think time, HTTP server time and network time.

Page load time threshold

The slow page threshold for the software service. This is the page load time threshold. If the page loads in a time longer than the threshold, it is considered to be a slow page. The value of the threshold can be set individually on per-software service basis.

Pages

The number of Web pages loaded.

Pages breakdown

The breakdown of all the pages into slow and fast pages.

Page size

The page size expressed in bytes.

Pages with HTTP errors (4xx 5xx)

The number of pages returned with a 4xx or a 5xx error.

Percentage of aborts

The percentage of page loads manually stopped by the user by either clicking the Stop or Refresh buttons or selecting another URL.

Percentage of slow pages

The percentage of pages that took longer to load (had longer Page load time) than their predefined threshold value (by default 8 seconds).

Percentage of slow pages due to DC

The percentage of pages that took longer to load (had longer Page load time) than their predefined threshold value (by default 8 seconds), where the cause of the long load time was data center.

Person-hours lost

In Vantage Analysis Server, the total monitoring time clients waited for pages to load due to bad service availability and bad application performance In Advanced Web Diagnostics Server, the total time clients waited for pages to load due to bad application performance, that is, the total monitoring time during which page load time exceeded the predefined threshold. Note that this is not a sum of whole monitoring intervals, but only those intervals' portions during which problems occurred.

Redirect time

The average amount of time that was spent between time when a user went to a particular URL and time this user was redirected to another URL and issued a request to that new URL. The difference between Redirect Time and HTTP Redirect Time is that the former counts all operations while the latter refers only to those operations, for which redirection actually took place.

Request body size

Size of the HTTP request body (sum for page hits).

Request cookie size

Length of the Cookie field content, including separators (sum for page hits).

Request header size

Average length of the HTTP request header (sum for page hits).

Request size

The number of bytes sent by a client as an HTTP request for the hit (sum for page hits).

Request time

The time it took the client to send the HTTP request to the server (for example, by means of an HTTP GET or HTTP POST). Note that this time includes TCP connection setup time and SSL session setup time (if any). It starts when the client starts the TCP session on the server and ends when the server receives the whole request. Sometimes a page is slow because of a big request rather than due to a large response.

Response cookie size

Length of the Set-Cookie field content, including separators (sum for page hits).

Response download time

The time it took to transfer the response from the server to the client.

Response header size

Average length of HTTP response header (sum for page hits).

Response throughput

Response throughput calculated as: Response size divided by Response download time.

Server bandwidth usage

The number of server bits per second.

Server bytes

The number of bytes sent by servers. The number includes headers.

Server loss rate

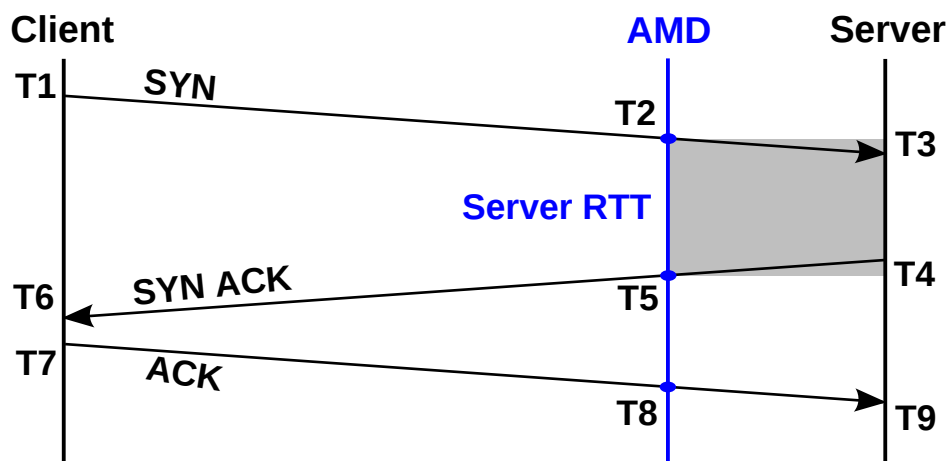
The percentage of the number of total packets sent from a server that were lost and needed to be retransmitted.

Server packets

The number of packets sent by the server(s).

Server RTT

The time it takes for a SYN packet (sent by a user) to travel from the AMD to a monitored server and back again.



Server think time

The time that elapsed between the moment the server received the request for the Base Page, and the time the server fully composed the response. Depending on the nature of the request, Application Servers in the Data Center may be involved to produce the content. In such a case, this additional time will be reflected in the Server Think Time metric.

Server time

The time it took the server to produce a response for the given request.

Slow pages

The number of slow page loads due to the given reason.

Slow pages (client delays)

The number of slow pages due to client delays.

Slow pages (data center)

The number of slow pages due to data center delays.

Slow pages (multiple reasons)

Slow pages due to multiple reasons are slow pages for which a reason could not be determined. Note that the various categories of errors form non-intersecting sets, that is, the pages listed as being slow for multiple reasons do not appear in the statistics for single reasons, like those that are slow due to page design or due to client delays.

Slow pages (network)

The number of slow pages due to network problems.

Slow pages (network - latency)

The number of slow pages due to network latency (high RTT).

Slow pages (network - loss rate)

The number of slow pages due to loss rate.

Slow pages (network - other reasons)

The number of slow pages due to other network reasons.

Slow pages (network - request time)

The number of slow pages due to request time.

Slow pages (page design)

The number of slow pages due to page design.

Slow pages (page design - # of components)

The number of slow pages due to high number of page components.

Slow pages (page design - page size)

The number of slow pages due to page size.

SSL conn. setup per session

The time it took to establish an SSL connection between the client and the server.

Stdv number of request cookies

Standard deviation of the number of the key-value pairs in the Cookie field of the HTTP request header.

Stdv number of response cookies

Standard deviation of the number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Stdv redirect time

The standard time of the introductory HTTP redirect.

Stdv request body size

The standard deviation of the size of the HTTP request body (sum for page hits).

Stdv request cookie size

Standard deviation of the length of the Cookie field content, including separators (sum for page hits).

Stdv request header size

Standard length of the HTTP request header (sum for page hits).

Stdv response cookie size

Standard deviation of the length of the Set-Cookie field content, including separators.

Stdv response header size

Standard length of HTTP response header (sum for page hits).

Time resolution

Time resolution.

Total bandwidth usage

The number of all transmitted bits (client + server) per second.

Total bytes

The number of all transmitted bytes (client + server).

Total packets

The number of all transmitted packets (client + server).

Transaction page begin

The length of time that elapsed between the beginning of a transaction and the start of the particular page load of that transaction.

Unique users

The number of users with unique network ID or login name, detected in the monitored traffic.

Transactions data view

Transactions data view dimensions

Agent

The name of the synthetic agent that loaded the HTTP page(s), for example, Keynote, Gomez, or Mercury. The name of the agent is determined from the User-agent field of the HTTP request and/or from agent user names or IP address configured on the server.

AMD UUID

UUID (Universal Unique Identifier) of the AMD that produced the data.

Analyzer group

An analyzer group name. For example: Web, XML or Oracle Forms.

Application

Name of the software service running on a given port.

Client area

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names, CIDR blocks or subnets. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client AS

The name of the client's BGP Autonomous System.

Client ASN

Autonomous System Number (ASN) of the client site.

Client city

Geographical data about the client site.

Client country

Geographical data about the client site.

Client geographical region

Geographical data about the client site.

Client group

The client's group, as manually defined in Vantage Analysis Server.

Client IP address

The IP address of the client.

Client IP address (internal)

Client IP extracted from the HTTP header on the AMD side.

Client region

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client site

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions, clients' BGP Autonomous System names, CIDR blocks or subnets. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client site description

Optional description of the client site.

Client site ID

In cases when sites are ASes, Client Site ID contains the AS number, which is also given in Client ASN. For manual sites, Client Site ID is identical to Client site, and contains

the site name as defined in your site configuration. Sites based on CIDR blocks or subnets are identified by IP addresses.

Client WINS name

The client's computer name resolved by a WINS server

Correlation ID

A unique numerical identifier of an XML/SOAP message, assigned by the AMD, used for linking messages into one transaction in the Transactions, Page Analysis and Page Elements data views.

Protocol

IP protocol name.

Time

The time stamp of the data presented on the report.

Traffic type

The type of client traffic: real or synthetic, that is, generated by a synthetic agent.

Transaction begin time

In addition to the measurement (AMD batch of data) time, each page is separately stamped with its actual time of occurrence, with 1-millisecond granularity. This dimension can be used to list pages one by one, in the order of occurrence, to reflect exactly the client or server activity, page by page.

Transaction ID

A unique identifier of a transaction to which this page belongs. A transaction is a sequence of page loads used for accomplishing a particular action, for example on-line travel or appointment booking, on-line shopping etc.

Transaction name

The name of the transaction.

Transaction status

HTTP transaction status or transaction error code: positive values represent HTTP status of the last page of the transaction, negative values give transaction error codes: -2 for page timeout or transaction timeout; -3 for transaction abort; -4 for lost pages: less pages belonging to the transaction are read than should be read according to the transaction definition; -5 other error.

User name

Client's name determined from HTTP cookie (requires configuration on AMD), HTTP authentication header or static mapping.

Transactions data view metrics

50pc network time

The 50th percentile of the network time.

50pc server time

The 50th percentile of the server time.

50pc transaction time

The 50th percentile of transaction time.

80pc network time

The 80th percentile of the network time.

80pc server time

The 80th percentile of the server time.

80pc transaction time

The 80th percentile of transaction time.

85pc network time

The 85th percentile of the network time.

85pc server time

The 85th percentile of the server time.

85pc transaction time

The 85th percentile of transaction time.

90pc network time

The 90th percentile of the network time.

90pc server time

The 90th percentile of the server time.

90pc transaction time

The 90th percentile of transaction time.

95pc network time

The 95th percentile of the network time.

95pc server time

The 95th percentile of the server time.

95pc transaction time

The 95th percentile of transaction time.

99pc network time

The 99th percentile of the network time.

99pc server time

The 99th percentile of the server time.

99pc transaction time

The 99th percentile of transaction time.

Application processing time

The average time spent by software service on operation processing.

Avg client bytes

The average number of bytes sent by the client(s). Note that this includes headers.

Avg client packets

The average number of packets sent by the client.

Avg server bytes

The average number of bytes sent by servers. Note that this includes headers.

Avg server packets

The average number of packets sent by the server.

Avg total bytes

The average number of all transmitted bytes (upstream + downstream).

Avg total packets

The average number of all transmitted packets (client + server).

Client bytes

The number of bytes sent by the client(s). Note that this includes headers.

Client loss rate

The percentage of total packets sent by a client that were lost (due to network congestion, router low queue capacity or other reasons) and needed to be retransmitted.

Client packets

The number of packets sent by the client(s).

Client response time

The average time spent by client side on transaction processing.

Client RTT

Client RTT is the time it takes for a SYN packet (sent by a server) to travel from the AMD to the user and back again.

Client time

Time during the page load when there is no network or server activity related to the page. It is assumed that Idle time is caused by the user's browser not sending requests because user's PC is busy.

Custom metric (1) (avg) (default name)

The average value of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (1)(cnt) (default name)

The number of occurrences of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (1) (sum) (default name)

The sum of all values of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (2) (avg) (default name)

The average value of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (2)(cnt) (default name)

The number of occurrences of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (2) (sum) (default name)

The sum of all values of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (3) (avg) (default name)

The average value of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (3)(cnt) (default name)

The number of occurrences of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (3) (sum) (default name)

The sum of all values of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (4) (avg) (default name)

The average value of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (4)(cnt) (default name)

The number of occurrences of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (4) (sum) (default name)

The sum of all values of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (5) (avg) (default name)

The average value of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Custom metric (5)(cnt) (default name)

The number of occurrences of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Custom metric (5) (sum) (default name)

The sum of all values of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Duplicated pages

The total number of duplicated pages (XML/SOAP only).

End-to-end RTT

The time it takes for a SYN packet to travel from the client to a monitored server and back again.

Failures (client HTTP errors)

The number of transaction requests that did not become successful transactions due to client HTTP errors: HTTP code between 400 and 499.

Failures (no server response)

The number of transaction requests that did not become transactions due to no server response.

Failures (server HTTP errors)

The number of transaction requests that did not become successful transactions due to server HTTP errors: HTTP code greater than or equal to 500.

Failures (transaction abandonments)

The number of transaction requests that did not become successful transactions due to the timeout for the request. By default, the timeout is set to 360 seconds, but it can be separately customized for each transaction request. The user left the Web server on which the transaction requests was being performed and started browsing a different Web site.

Failures (transaction aborts)

The number of transaction requests that did not become successful transactions due to user's abort. During the completion of the request, the user provided incorrect information. So, the user left the transaction request being performed and started over with a new request.

Failures (transaction incomplete)

The total number of transaction requests that became incomplete transactions, that is, transactions for which monitored traffic did not match the first step(s) in the transaction definition.

Hit errors

The number of hit errors in transaction pages (where hit errors cause HTTP status to be less or equal than 599 and greater than or equal to 400)

Hits

The number of subcomponents of error-free operations or transactions. Note that this metric is recorded at the time when the monitored transactions are closed.

HTTP application responses

The number of HTTP application responses of types 1-5, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 1 (default name)

The number of HTTP application responses of type 1, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 2 (default name)

The number of HTTP application responses of type 2, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 3 (default name)

The number of HTTP application responses of type 3, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 4 (default name)

The number of HTTP application responses of type 4, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 5 (default name)

The number of HTTP application responses of type 5, observed for the HTTP- or HTTPS-based software service.

Network time

The time the Internet (between the user and the server) takes to deliver requests to the server and to deliver page information back to the user. In other words, Network time is the portion of transaction time that is due to the delivery time on the network/Internet. Also provided are minimum and maximum values.

Nominal transaction steps

Average number of transaction steps as configured. This can be smaller than the number of steps actually taken, if repetitions are allowed.

Page aborts

The number of transaction pages aborted by the client.

Page idle time

The sum of the idle time for all pages used in the transaction.

Pages

The number of Web pages loaded.

Percentage of slow transactions

The percentage of slow transactions in all detected transactions.

Server bytes

The number of bytes sent by servers. The number includes headers.

Server loss rate

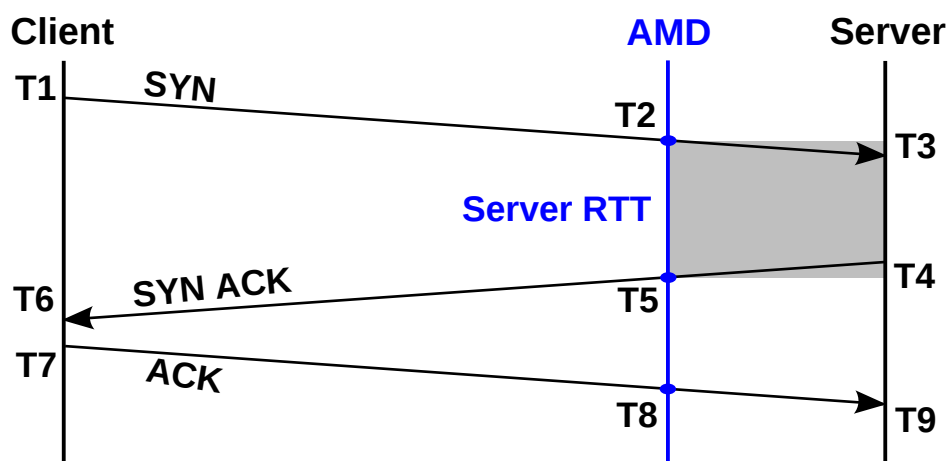
The percentage of the number of total packets sent from a server that were lost and needed to be retransmitted.

Server packets

The number of packets sent by the server(s).

Server RTT

The time it takes for a SYN packet (sent by a user) to travel from the AMD to a monitored server and back again.

**Server time**

This is the sum of server time for all the pages that belong to the transaction.

Slow transactions

The number of transactions detected as slow.

Slow transaction threshold

Average slow transaction threshold time, as per configuration

Time resolution

The value of the time resolution for the given report. While being of constant value for a particular report, this metric will be present or absent, for different values of the time dimension, thus showing user activity in time. It can also provide useful information when compared - on the same graph - with aggregate time metrics.

Total bytes

The number of all transmitted bytes (client + server).

Total packets

The number of all transmitted packets (client + server).

Transaction requests

The number of all transactions request, both successful transactions and transactions with errors.

Transactions

The number of transactions executed.

Transactions breakdown

The breakdown of all the transactions into slow and fast transactions.

Transaction steps

The number of transaction steps.

Transaction time

The time it took to complete the transaction. Note that a transaction can consist of many HTTP pages.

Transaction time breakdown

The transaction time breakdown into server time and network time.

Unique clients

The number of clients with the unique network ID or login name, detected in the monitored traffic.

Page Elements data view

Page Elements data view dimensions

AMD UUID

UUID (Universal Unique Identifier) of the AMD that produced the data.

Analyzer group

An analyzer group name. For example: Web, XML or Oracle Forms.

Browser agent

The name of the agent, as reported by a Web browser.

Browser name

The name of the Web browser, as reported by a browser software.

Browser OS

The name of the OS hosting the Web browser. Example: Windows 5.0.

Client area

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names, CIDR blocks or subnets. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client AS

The name of the client's BGP Autonomous System.

Client ASN

Autonomous System Number (ASN) of the client site

Client city

Geographical data about the client site.

Client country

Geographical data about the client site.

Client geographical region

Geographical data about the client site.

Client group

The client's group, as manually defined in Vantage Analysis Server.

Client IP address

The IP address of the client.

Client IP address (internal)

Client IP extracted from the HTTP header on the AMD side.

Client port

The TCP port number of the client machine on which the software service is running.

Client region

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client site

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions, clients' BGP Autonomous System names, CIDR blocks or subnets. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Client site description

Optional description of the client site.

Client site ID

In cases when sites are ASes, Client Site ID contains the AS number, which is also given in Client ASN. For manual sites, Client Site ID is identical to Client site, and contains the site name as defined in your site configuration. Sites based on CIDR blocks or subnets are identified by IP addresses.

Client WINS name

The client's computer name resolved by a WINS server

Component URL

This is an actual URL intercepted from the HTTP request.

Content type

The string determining the type of the content served by the HTTP server. Example: text/html, text/xml, application/pdf or image/gif.

Correlation ID

A unique numerical identifier of an XML/SOAP message, assigned by the AMD, used for linking messages into one transaction in the Transactions, Page Analysis and Page Elements data views.

Hit begin time

In addition to the measurement (AMD batch of data) time, each page is separately stamped with its actual time of occurrence, with 1-millisecond granularity. This dimension can be used to list pages one by one, in the order of occurrence, to reflect exactly the client or server activity, page by page.

Hit ID

An ID of the HTTP hit, being part of the page load. This is a unique numerical identifier assigned by the AMD, used for linking page hits into a page in the Page Elements data view. Note that when selecting the Hit ID as a dimension, you cannot specify more than one value in the Condition/Negate field.

Hit status

The textual description of the status of the hit.

HTTP application response messages

The messages of categories 1-5, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category1 (default name)

The messages of category 1, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category2 (default name)

The messages of category 2, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category3 (default name)

The messages of category 3, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category4 (default name)

The messages of category 4, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP application response messages - category5 (default name)

The messages of category 5, communicated to the user by means of HTML pages if a given pattern is found or is missing in an HTTP data stream.

HTTP response status

Response status as read from the server response header. DMI also displays status name as a result of the query. Value -1 means 'No response header', other values as from the definition of the HTTP protocol.

Operation ID

A unique numerical identifier of a page load assigned by the AMD, used for linking page hits into one page on the Page Elements data view. Note that when selecting the Operation ID as a dimension, you cannot specify more than one value in the Condition/Negate field.

Operation status

The status of the operation to which the hit belongs. It indicates if the operation was considered a regular page load.

Page name

Alias for the page URL set on AMD's side or page URL itself if alias is undefined.

Page URL

For HTTP, this is the URL of the base page to which the hit belongs. For other analyzers this can be a query or an operation type. Page URL is ascertained by the AMD, based on referrer, timing relations between hits and per-transaction monitoring configured on the AMD. This dimension can assume values of a particular URL - if this URL is monitored. Note: The visibility of this dimension on reports depends on whether another dimension, related to servers - e.g. server IP or server DNS - has been used when formulating the query. Compare "Page URL (incl. whole)".

Parent reporting group

An application that acts as a "container" for a specific transaction.

POST data

The data returned (pre-formatted) by the Web server in the POST stream.

POST data (raw)

The raw data returned by the Web server in the POST stream.

Query text (raw)

A full representation of the query, not preprocessed by the AMD.

Reporting group

Reporting group is a universal container that can accommodate software services, servers, URLs or any combination of these. Reporting groups can contain software services of every type. Advanced Web Diagnostics Server can import reporting group configuration from Vantage Analysis Server.

Request cookie

The content of the Cookie field from the HTTP request header.

Request header

The content of the HTTP request header.

Request method

HTTP request type: GET or POST.

Request parameters (from URL)

The parameters string of a GET/POST request containing all the request parameters and their values, as they appear in the URL.

Response cookie

The content of the Set-Cookie field from the HTTP response header.

Response header

The content of the HTTP response header.

Response status

It indicates if there was a valid response header or not. It assumes the following values: 'Response OK' and 'No response header'.

Server area

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and optionally on clients' BGP Autonomous System names, CIDR blocks or subnets. Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Server AS

The name of the Autonomous System of the server site.

Server ASN

Autonomous System Number (ASN) of the server site.

Server city

Geographical data about the server site.

Server country

Geographical data about the server site.

Server DNS name

The name of the server resolved by a DNS server.

Server geographical region

Geographical data about the server site.

Server host name

The name of the server, as read from the Host field of the packet header.

Server IP address

The IP address of the server.

Server port

The TCP port number on a server that hosts a software service.

Server region

Sites, areas and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' BGP Autonomous System names.

Sites are the smallest groupings of clients and servers. Areas are composed of sites. Regions are composed of areas.

Server site

Sites, areas, and regions define a logical grouping of clients and servers into a hierarchy. They are based on manual definitions and/or on clients' Autonomous System names. Sites are the smallest logical structures that comprise of clients and servers. Areas are composed of sites, while regions are composed of areas.

Server site description

Optional description of the server site.

Server site ID

In cases when sites are ASes, Server site ID contains the AS number, which is also given in Server ASN. For manual sites, Server site ID is identical to Server site, and contains the site name as defined in your site configuration. Sites based on CIDR blocks or subnets are identified by IP addresses.

Software service

Software service name, where by a software service we understand a service implemented by a specific piece of software, offered on a TCP or UDP port of one or more servers and identified by a particular TCP port number.

Time

The time stamp of the data presented on the report.

User name

Client's name determined from HTTP cookie (requires configuration on AMD), HTTP authentication header or static mapping.

Page Elements data view metrics

Client bytes

The number of bytes sent by the client(s). Note that this includes headers.

Client loss rate

The percentage of total packets sent by a client that were lost (due to network congestion, router low queue capacity or other reasons) and needed to be retransmitted.

Client packets

The number of packets sent by the client(s).

Client RTT

Client RTT is the time it takes for a SYN packet (sent by a server) to travel from the AMD to the user and back again.

Client zero window size events

Client sets this in TCP header when it wants the other side to slow down with data transmission because it cannot keep up with the transmission speed. Indicates the client's computer is busy with other tasks.

Component request begin

The difference between the hit begin time and the page begin time.

Custom metric (1) (avg) (default name)

The average value of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (1)(cnt) (default name)

The number of occurrences of user-defined metrics in category 1 observed in the HTTP or XML traffic..

Custom metric (1) (sum) (default name)

The sum of all values of user-defined metrics in category 1 observed in the HTTP or XML traffic.

Custom metric (2) (avg) (default name)

The average value of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (2)(cnt) (default name)

The number of occurrences of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (2) (sum) (default name)

The sum of all values of user-defined metrics in category 2 observed in the HTTP or XML traffic.

Custom metric (3) (avg) (default name)

The average value of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (3)(cnt) (default name)

The number of occurrences of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (3) (sum) (default name)

The sum of all values of user-defined metrics in category 3 observed in the HTTP or XML traffic.

Custom metric (4) (avg) (default name)

The average value of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (4)(cnt) (default name)

The number of occurrences of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (4) (sum) (default name)

The sum of all values of user-defined metrics in category 4 observed in the HTTP or XML traffic.

Custom metric (5)(cnt) (default name)

The number of occurrences of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Custom metric (5) (sum) (default name)

The sum of all values of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Custom metric (51) (avg) (default name)

The average value of user-defined metrics in category 5 observed in the HTTP or XML traffic.

Hits

The number of subcomponents of error-free operations or transactions. Note that this metric is recorded at the time when the monitored transactions are closed.

HTTP application responses

The number of HTTP application responses of types 1-5, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 1 (default name)

The number of HTTP application responses of type 1, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 2 (default name)

The number of HTTP application responses of type 2, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 3 (default name)

The number of HTTP application responses of type 3, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 4 (default name)

The number of HTTP application responses of type 4, observed for the HTTP- or HTTPS-based software service.

HTTP application responses – category 5 (default name)

The number of HTTP application responses of type 5, observed for the HTTP- or HTTPS-based software service.

Max client RTT

The maximum of the client RTT.

Max component request begin

The maximum of the component request begin time.

Max number of request cookies

Maximum number of the key-value pairs in the Cookie field of the HTTP request header.

Max number of response cookies

Maximum number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Max request body size

Maximum of the size of the HTTP request body (sum for page hits).

Max request cookie size

Maximum length of the Cookie field content, including separators.

Max request header size

Maximum length of the HTTP request header.

Max request size

The maximum of the request size.

Max request time

The maximum of the request time.

Max response cookie size

Maximum length of the Set-Cookie field content, including separators (sum for page hits).

Max response download time

The maximum of the response download time.

Max response header size

Maximum length of HTTP response header.

Max response size

The maximum of the response size.

Max server RTT

The maximum of the server RTT.

Max server time

The maximum server time.

Max SSL conn. setup per session

The maximum of the SSL connection setup per session.

Min client RTT

The minimum of the client RTT.

Min component request begin

The minimum of the component request begin time.

Min number of request cookies

Minimum number of the key-value pairs in the Cookie field of the HTTP request header.

Min number of response cookies

Minimum number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Min request body size

Minimum of the size of the HTTP request body.

Min request cookie size

Minimum length of the Cookie field content, including separators.

Min request header size

Minimum length of the HTTP request header.

Min request size

The minimum of the request size.

Min request time

The minimum of the request time.

Min response cookie size

Minimum length of the Set-Cookie field content, including separators (sum for page hits).

Min response download time

The minimum of the response download time.

Min response header size

Minimum length of HTTP response header.

Min response size

The minimum of the response size.

Min server RTT

Minimum server RTT.

Min server time

The minimum of the server time.

Min SSL conn. setup per session

Minimum SSL connection setup per session.

Number of request cookies

The number of key-value pairs in the Cookie field of the HTTP request header.

Number of response cookies

The number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Request body size

Size of the HTTP request body.

Request cookie size

Length of the Cookie field content, including separators.

Request header size

Average length of the HTTP request header.

Request size

The number of bytes sent by a client as an HTTP request for the hit.

Request time

The time it took the client to send the HTTP request to the server (for example, by means of an HTTP GET or HTTP POST). Note that this time includes TCP connection setup time and SSL session setup time (if any). It starts when the client starts the TCP session on the server and ends when the server receives the whole request. Sometimes a page is slow because of a big request rather than due to a large response.

Response cookie size

Length of the Set-Cookie field content, including separators.

Response download time

The time it took to transfer the response from the server to the client.

Response header size

Average length of HTTP response header.

Response size

The number of bytes sent by the server in response to an HTTP request.

Server bytes

The number of bytes sent by servers. The number includes headers.

Server loss rate

The percentage of the number of total packets sent from a server that were lost and needed to be retransmitted.

Server packets

The number of packets sent by the server(s).

Server RTT

The time it takes for a SYN packet (sent by a user) to travel from the AMD to a monitored server and back again.

Server time

The time it took the server to produce a response for the given request.

SSL conn. setup per session

The time it took to establish an SSL connection between the client and the server.

Stdv client RTT

The standard deviation of the client RTT.

Stdv component request begin

The standard deviation of the component request begin time.

Stdv number of request cookies

Standard deviation of the number of the key-value pairs in the Cookie field of the HTTP request header.

Stdv number of response cookies

Standard deviation of the number of the key-value pairs in the Set-Cookie field of the HTTP response header.

Stdv request body size

The standard deviation of the size of the HTTP request body.

Stdv request cookie size

Standard deviation of the length of the Cookie field content, including separators.

Stdv request header size

Standard length of the HTTP request header.

Stdv request size

The standard deviation of the HTTP request size.

Stdv request time

The standard deviation of the request time.

Stdv response cookie size

Standard deviation of the length of the Set-Cookie field content, including separators.

Stdv response download time

The standard deviation of the response download time.

Stdv response header size

Standard length of HTTP response header.

Stdv response size

The standard deviation of the response size.

Stdv server RTT

The standard deviation of the server RTT.

Stdv server time

The standard deviation of the server time.

Stdv SSL conn. setup per session

The standard deviation of the SSL connection setup per session.

Time resolution

Time resolution.

Total bytes

The number of all transmitted bytes (client + server).

Total packets

The number of all transmitted packets (client + server).

Unique client IP addresses

The number of unique IP addresses of the client. When clients are aggregated to so-called aggregation blocks, only the most active IP addresses per reported entity are kept in the database.

Unique users

The number of users with unique network ID or login name, detected in the monitored traffic.

APPENDIX C

Classification of Aborts

HTML pages that did not load properly or the page load process did not complete generate “events” that are called aborts. Such events are then associated with the page, for which the “event” has been detected.

Based on the transaction status we can distinguish two general categories of aborts:

- Transaction for which there was no HTTP server response detected
- Transaction aborted after the HTTP server responded with an HTTP header

Each of these types of aborts may occur in different circumstances, identified as *Break*, *Client abort*, *Dead hit* or *Error*. In addition, each hit, for which an abort was detected, can be a regular hit or it can be a stand-alone hit (with no HTML page context). Each of the categories listed below is divided further into those two subcategories. The following table summarizes conditions when particular aborts are encountered.

Table 3. Classification of aborts

Type	No Server Response (AMD did not detect any valid HTTP response header)	Transaction Abort (happened after the server has already sent a valid HTTP response header to the client)
Break	<p>The server sends a packet with the TCP RST flag to a client—instead of any response. The most probable cause is one of the following:</p> <ul style="list-style-type: none">• The server application has found that the client has been idle for too long (not finishing the request despite the server acknowledgment to all client packets) and the transaction should be closed• The TCP session has been idle for too long. (Timeout is 5 minutes by default.	<p>First, a server sends some response data. Then it sends a packet with TCP RST flag to the client. The most probable cause is that the server application has found reasons to immediately close the transaction</p> <p>OR</p> <p>the TCP session has been idle for too long.</p>

Type	No Server Response (AMD did not detect any valid HTTP response header)	Transaction Abort (happened after the server has already sent a valid HTTP response header to the client)
	<p>However, this condition is checked only during sample generation on the AMD, so actual dead time may vary from 5 to 10 minutes if samples are generated every 5 minutes.)</p>	
Client Abort	A client sent a TCP RST packet to the server before the server ever responded to the request. The TCP session was closed.	A client sent a TCP RST packet to the server after receiving some data from the server. Perhaps it has already received what it was waiting for and does not need the rest of response.
Dead	<p>A gap of 45 seconds or more between packets (server packets or client packets). A 45-second timeout is applied only during sample generation. This means that the timeout may vary from 45 seconds to 5 minutes if samples are generated every 5 minutes.</p> <p>Note that this condition is checked for HTTP hits, while "Break" is checked for the TCP session.</p>	The same as Dead, No Server Response, but we have seen a response header.
Error	<p>There was a server response, but without an HTTP header and the response occurred 60 or more seconds after receiving request or client sent new request before consuming the response</p> <p>OR</p> <p>the transaction was idle for too long</p> <p>OR</p> <p>the client was sending the request too slowly (there was a gap of more than 10 seconds between consecutive TCP packets).</p>	The same as Error, No Server Response, but we have seen a response header.

Index

A

Advanced Web Diagnostics Server, See AWDS

alarm

- canceling 67
- defining 74
- delivery 67, 74
- event 69
- mechanism 67
- message content 78
- notification 69
 - e-mail 76
 - sending 76
- raising 67
- raising and cancelling conditions 67
- repeating 67, 69
- SNMP traps for alarm notifications 76
- states 69
- types 73

analyzer

- supported by AWDS 79

AWDS 11

- release information 11
- supported protocols 79

B

browser

- configuring 12
- localization 14
- versions supported 12

C

configuration

- browser 12

D

Data Center Infrastructure Performance 37

decode

- supported by AWDS 79

DMI

- home page 21
- report 21
 - Business Transaction Monitoring 59, 61
 - Portal Applications 52
 - Transaction Agent 60
 - Transaction Monitoring 59
 - Transaction Users 60
 - User Path Through Site 52

H

HTTP 79

- application responses 56

I

international features support

- character encoding 14
- East Asian languages 18
- localized browser 14
- localized reports 18
- localized server 14, 17, 18
- user settings 18

L

languages

- East Asian 18

localization

- browser 14
- character encoding 14
- server 14

O

operation log 65

Oracle E-Business Suite 79

Index

Oracle Forms 79

AWDS 43

support 43

P

Page Analysis data view

dimensions and metrics 81

Page Elements data view

dimensions and metrics 108

protocol

analyzer 79

supported by AWDS 79

Q

query

query log 64

R

release information

AWDS 11

repeating alarm notification 67, 69

report

application response log 58

application response messages 57

application responses 56

Application-URLs 28

AWDS 23

Continuous Improvement Gauge 42

Customer Care - Affected User Performance 40

Hit Details 36, 55

language settings 18

load sequence 64

LOB Details 25

Orphaned Redirects 51

Slow Page Cause Breakdown 63

Slow Page Load Sequence 36

Slow Page Loads 30

report (*continued*)

stepchart 64

User Path Through Site 52

report server

supported browsers 12

report tree 22

S

Siebel

CRM 79

Slow Page Cause Breakdown 63

slow pages

classification 32

SNMP

traps for alarm notifications 76

software service

performance 63

SSL 79

system requirements

supported browsers 12

T

transaction

log 65

Transactions data view

dimensions and metrics 100

traps

SNMP 76

U

User Activity

User report

load sequence 64

V

version numbers 11