



PDF Report Templates Handbook

Version 5.0
English Edition



Table of Contents

1	Preliminary Note.....	3
2	Overview and First Steps	3
2.1	Additional Hints	5
2.1.1	Report Header.....	5
2.1.2	CSS Files	6
2.1.3	Special Characters	6
2.1.4	The XXXXXX.gif Image	6
2.1.5	Debugging Malformed Report Templates	7
3	Placeholders / Simple Server Side Includes	8
4	Using Report Templates from the PdfReport Utility	9
4.1	Automated Filling In of Comment Fields	10
5	Manufacturer	12

1 Preliminary Note

The design of customer-specific PDF Reports templates is quite difficult and usually only done by the Proxy Sniffer manufacturer itself, or by a Proxy Sniffer general agency, based on a special consulting contract. This means that such tasks are not covered by the standard Proxy Sniffer support terms.

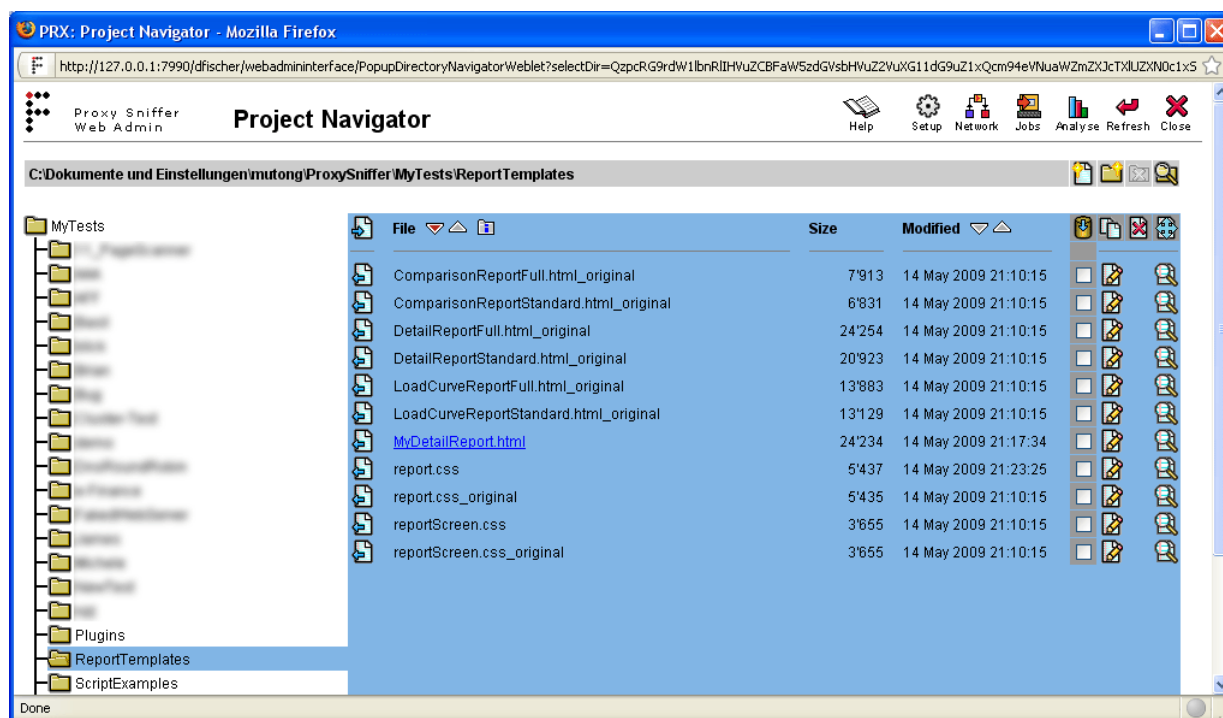
Nevertheless, you can make **modifications** yourself based **on the existing pre-defined report templates**. The most frequent use cases are that some functionality of a predefined report template needs to be deleted or rearranged, or that the layout of the PDF report must be adjusted to meet your own requirements.

Please note the following important points:

- Making or modifying PDF report templates requires expert-level knowledge of XHTML and CSS.
- We cannot give you support for defining own report templates – based on the “standard support option”. A special, additional consulting contract is required if you wish to receive support for customized PDF report templates.
- **The documentation in this handbook is currently incomplete.**

2 Overview and First Steps

Starting from Proxy Sniffer version 4.2-P, a copy of all existing pre-defined report templates is placed by the Windows and Mac OS X installation kit inside the **ReportTemplates** folder in the **MyTests** directory of the Project Navigator. On other Unix-like systems (Linux, Solaris, BSD ...), you must create the **ReportTemplates** folder manually, and then copy the predefined report templates from a Windows or Mac OS X system to this folder.



Next, you can make a copy of an existing pre-defined report template by giving it a new filename which ends with ***.html**.

In the following example, a copy of the file **DetailReportFull.html_original** has been stored under the name **MyDetailReport.html**.

Now you can edit the file **MyDetailReport.html** using a normal text editor. The modifications are marked in **red color text**:

```

<!--
#reportType detailResult
#reportLabel My Detail Report
#reportAbstract Customized detail report. A4 landscape format.
#reportSortPosition 1
-->

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title><!-- @@dfischer.report.DetailReportTitleSSSI() --></title>
<link rel="stylesheet" type="text/css" media="print" href="report.css" />
<link rel="stylesheet" type="text/css" media="screen" href="reportScreen.css" />
</head>
<body>
<!-- START OF REPORT - BEFORE FIRST PAGE -->

<!-- set footer for first page -->
<div id="footer_firstPage" style=""><!-- @@dfischer.report.FirstPageFooterSSSI() --></div>

<!-- set header and footer for all other pages -->
<div id="header" style=""><!-- @@dfischer.report.DetailReportTitleSSSI() --></div>
<div id="footer" style="">Page <span id="pagenumber"/> of <span id="pagecount"/></div>

<!-- reset chapter numbers of document -->
<!-- @@dfischer.report.ReportTitleLevel_Reset_SSSI() -->

<!-- enable user comments - start marker -->
<!-- @@dfischer.report.ReportUserCommentStartSSSI() -->

<!-- COVER SHEET / FRONT PAGE -->
<!-- ===== -->

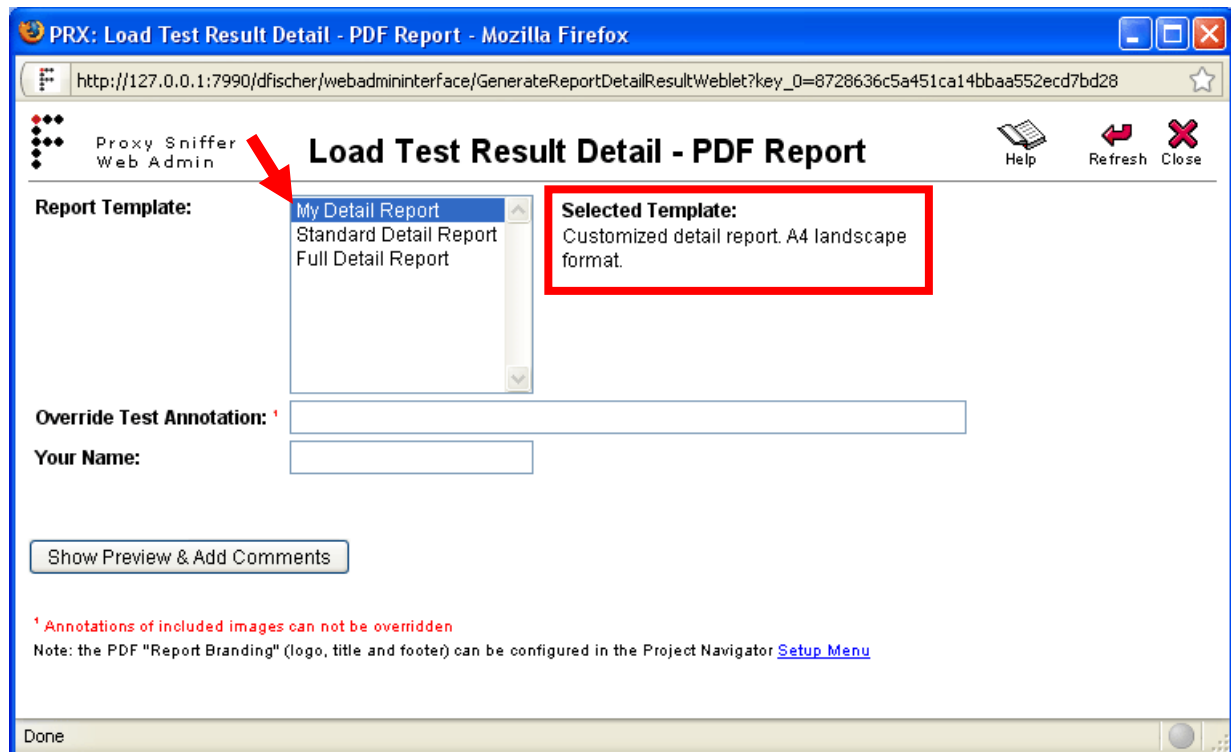
<!-- display report title -->
<div class="title_1"><!-- @@dfischer.report.DetailReportTitleSSSI() --></div>

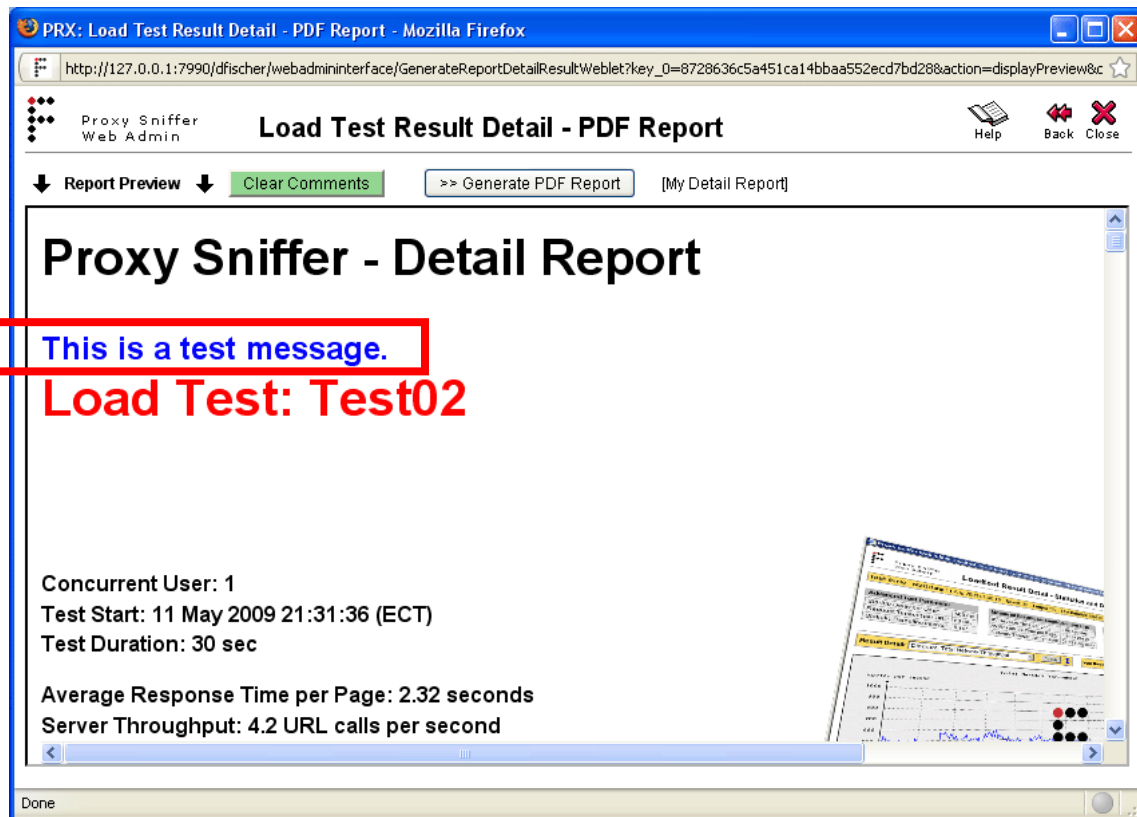
<br />&#160;<br />
<div style="font-size:18pt; font-weight:bold; color:blue;">This is a test message.</div>

<!-- display load test program name -->
<div class="title_1_red">Load Test: <!-- @@dfischer.report.DetailReportLoadTestProgramNameSSSI() --></div>

<!-- display annotation -->
<div class="title_2_red"><!-- @@dfischer.report.DetailReportAnnotationSSSI("Annotation: ", "true") --></div>
    
```

After this, you can generate your customized PDF detail report:





2.1 Additional Hints

2.1.1 Report Header

Each report template must contain a report header, which is written as an XHTML comment, but interpreted by Proxy Sniffer. Example:

```
<!--
#reportType detailResult
#reportLabel Full Detail Report
#reportAbstract Full report inclusive all details. A4 landscape format.
#reportSortPosition 14
-->
```

All report header fields are mandatory:

- **reportType**: possible values are **detailResult**, or **comparisonResult**, or **loadCurveResult**. A “detailResult” report only contains data about one test result. A “comparisonResult” report allows the comparison of results of several test runs. A “loadCurveResult” report shows the load curves; this means that the same load test program was executed several times by using a different number of simulated users. Note: depending on the report type, different placeholders are available (see chapter 3 in this document).
- **reportLabel**: contains the name of the report template which is shown in the GUI, and also used by the **PdfReport** utility (see chapter 4 in this document).
- **reportAbstract**: contains an abstract describing the content of the report. The abstract is shown in the GUI.
- **reportSortPosition**: contains a hint (a positive number) about the order in which report templates of the same report type are listed in the GUI.

2.1.2 CSS Files

All pre-defined PDF report templates refer to two CSS files in the **ReportTemplates** folder which also can be modified or extended manually:

- **reportScreen.css** ➔ used when the preview of the PDF report is displayed in the GUI.
- **report.css** ➔ used when generating the PDF report.

In most cases, you need to make your changes in both CSS files.

2.1.3 Special Characters

All special characters such as umlauts – including also enforced space characters – must be written in **Unicode notation**. Using a plain special character or a character in HTML notation, inside a report template is not possible and will result in an error.

Examples:

Special Character	Wrong HTML Notation	Proper Unicode Notation
' ' (enforced space)	 	
©	©	©
ü	ü	ü

2.1.4 The XXXXXX.gif Image

The XXXXXX.gif image which is located in the **ReportTemplates** folder is transparent (no color) and has a width and a height of 1 pixel. This image is used to adjust empty spaces in report layouts and is often referenced in scaled format. Example:

```

```

2.1.5 Debugging Malformed Report Templates

The use of XHTML implies that a report template must be 100% free of XHTML syntax errors. Typically, if an error is included inside the template, the report preview will be displayed more or less correctly in the GUI, but the generation of the PDF report fails and a "500 Internal Server Error" is shown.

Because debugging malformed report templates can be very complicated, it is strongly recommended that you make only small changes in the template in a step-by-step manner, and that you test each small modification immediately by generating a new PDF report.

If an XHTML syntax error occurs during the generation of a PDF report, a corresponding error message is usually shown in the Proxy Sniffer Console. Example:

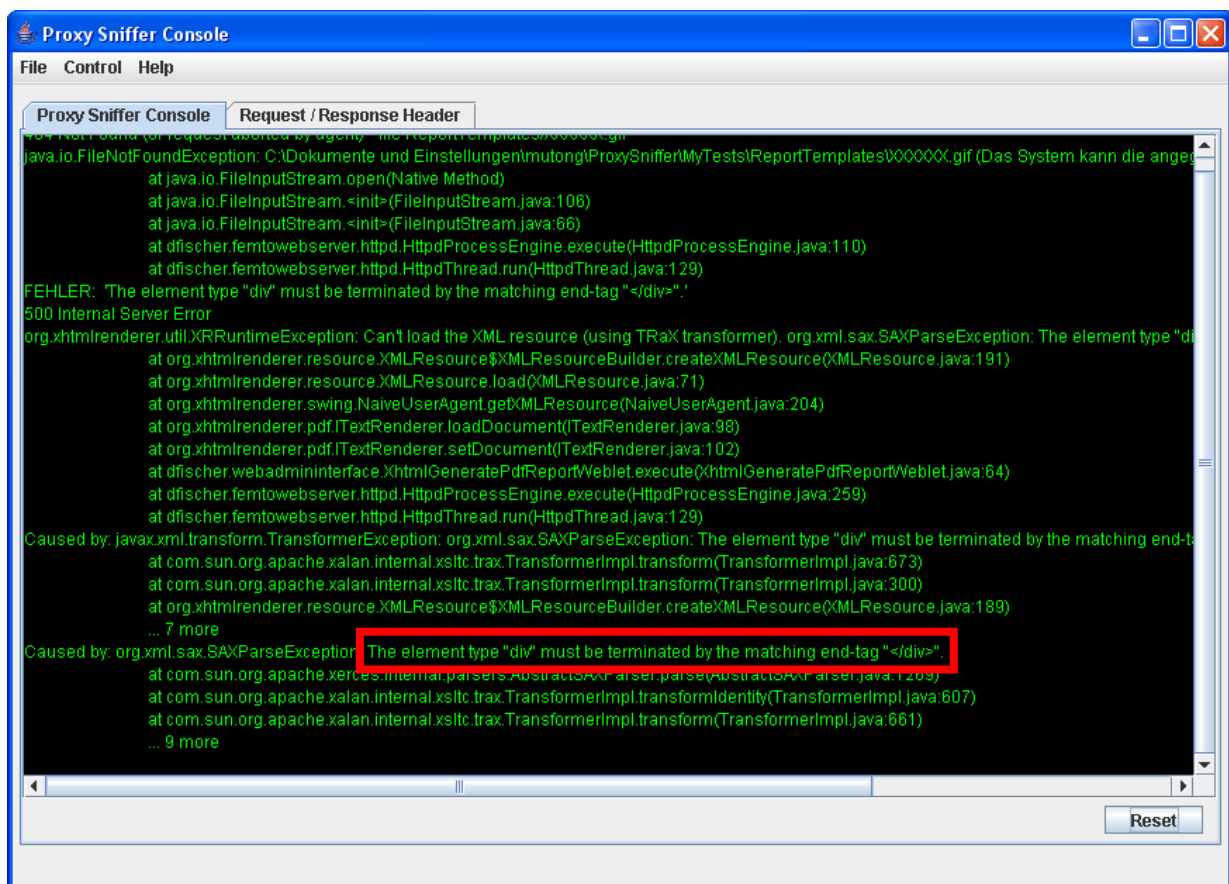
Error-free XHTML code fragment:

```
<div style="font-size:18pt; font-weight:bold; color:blue;">This is a test message.</div>
```

Malformed XHTML code fragment (opening angle bracket '<' absent in </div>):

```
<div style="font-size:18pt; font-weight:bold; color:blue;">This is a test message. /div>
```

Corresponding Error Message:



3 Placeholders / Simple Server Side Includes

All PDF report templates use placeholders as a core technique. These are also known as “Simple Server Side Includes”.

All placeholders begin with `<!-- @@dfischer.report.` and end with `-->`.

Example:

```
<!-- @@dfischer.report.DetailReportConcurrentUserSSSI() -->
```

The placeholder in this example shows the value of the number of concurrent users which have been simulated during load test execution.

The name of the placeholder is “DetailReportConcurrentUserSSSI” which means – because the name begins with “DetailReport ...” – that this placeholder can be only used in detail reports (See chapter 2.1.1, reportType = detailResult). “...SSSI” at the end of the name means that that this name refers to a placeholder (SSSI = Simple Server Side Include).

A placeholder can contain a single measured value (formatted text), as well as a measuring diagram (image), a entire table of measuring results, a document title (chapter or subchapter of the PDF report), or a “green” input field in which some user comments can be entered.

There is currently no documentation available about the placeholders; however, you can identify the usage of all supported placeholders in the corresponding ***Full.html_original** report templates. Updated documentation about placeholders will be published in future releases of Proxy Sniffer.

Hint: You may remove some placeholders from a report template to identify the content of a placeholder. Many placeholders also contain a hint about their functionality in the placeholder name.

4 Using Report Templates from the PdfReport Utility

The PdfReport utility (described in the “Application Reference Manual”, chapter 3.9) allows you to generate PDF reports from the command prompt of the operating system. This utility is normally used in combination with scripting the execution of load test programs (see “Application Reference Manual”, chapter 4).

All report templates – including your own customized templates – can be used from the PdfReport utility. The following example displays a list of all available templates:

```
C:\Documents and Settings\miller\ProxySniffer>java PdfReport listtemplates
```

```
Available Detail Report Templates:
```

- "My Detail Report"
- "Standard Detail Report"
- "Full Detail Report"

```
Available Comparison Report Templates:
```

- "Standard Comparison Report"
- "Full Comparison Report"

```
Available Load Curves Report Templates:
```

- "Standard Load Curves Report"
- "Full Load Curves Report"

A PDF report which uses a specific template can be generated by passing the option `-reportTemplate <name>` to the PdfReport utility. Example:

```
java PdfReport loadWithDetailReport  
  .\MyTests\Test01_20u.prxres -reportTemplate "My Detail Report"
```

4.1 Automated Filling In of Comment Fields

The PdfReport utility also supports the filling-in of customized text in any user comment field of the generated PDF reports (shown as green marked input text areas in the report preview) by passing one or more times the option `-comment:<name> <value>`.

To find the comment names supported by a report template, you can search inside the XHTML data of the template (*.html file) for all placeholders of

```
<!-- @@dfischer.report.ReportUserCommentAddSSSI(. . .) -->
```

Example:

```
<!-- CHAPTER: Test Scenario -->
<!-- ===== -->

<!-- main chapter title: Test Scenario -->
<div class="title_level_1"><!-- @@dfischer.report.ReportTitleLevel_1_SSSI("Test Scenario") --></div>

<!-- start of table around user comment for test scenario -->
<table border="0" cellpadding="0" cellspacing="0" style="border-collapse: collapse;">
<tr><td valign="top">

<!-- Initial Status -->
<table border="0" cellpadding="1" cellspacing="0" style="border-collapse: collapse; margin-bottom: 20px;">
<tr><td class="blackBorderNowrap" bgcolor="#81B5E5" colspan="2"><b>Initial Status</b></td></tr>
<tr><td class="blackBorderNowrap">Test Start Date:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.DetailReportT
<tr><td class="blackBorderNowrap">Load Test Program:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.DetailRepor
<tr><td class="blackBorderNowrap" valign="top">Load Source Host(s):</td><td class="blackBorderNowrap" valign="top"><!-- @
<tr><td class="blackBorderNowrap">Load Source OS:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.DetailReportLo
<tr><td class="blackBorderNowrap" valign="top">Target Host(s):</td><td class="blackBorderNowrap" valign="top"><!-- @dfis
<tr><td class="blackBorderNowrap">Applied HTTP Version:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.DetailRe
</table>

<!-- Test Input Parameter -->
<table border="0" cellpadding="1" cellspacing="0" style="border-collapse: collapse;">
<tr><td class="blackBorderNowrap" bgcolor="#81B5E5" colspan="2"><b>Test Input Parameter</b></td></tr>
<tr><td class="blackBorderNowrap">Concurrent User:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.DetailReportC
<tr><td class="blackBorderNowrap">Planned Test Duration:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.DetailR
<tr><td class="blackBorderNowrap">Planned Loops per User:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.Detail
<tr><td class="blackBorderNowrap">Startup Delay per User:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.Detail
<tr><td class="blackBorderNowrap">Request Timeout per URL:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.Detail
<tr><td class="blackBorderNowrap">Statistic Sampling Interval:</td><td class="blackBorderNowrap"><!-- @@dfischer.report.D
<tr><td class="blackBorderNowrap">Additional Sampling Rate per Web Page Call:</td><td class="blackBorderNowrap"><!-- @@df
<tr><td class="blackBorderNowrap">Additional Sampling Rate per URL Call:</td><td class="blackBorderNowrap"><!-- @@dfische
</table>

<!-- end of table around user comment for test scenario -->
</td>
<td></td>
<td valign="top"><!-- @@dfischer.report.ReportUserCommentAddSSSI("Comment:", "userCommentTestScenario",100,20) --></td>
</tr></table>

<!-- User Input Fields -->
<!-- @@dfischer.report.DetailReportDisplayUserInputFields("User Input Fields", "#81B5E5", "20px", "blackBorder") -->

<!-- Warnings -->
<!-- @@dfischer.report.DetailReportDisplayWarningsSSSI("Warnings", "#BBBBBB", "20px") -->

<!-- Real Time Comments -->
```

In this example the comment name is `"userCommentTestScenario"`.

The PDF report can then generated with the following command, which also fills in the corresponding comment:

```
java PdfReport loadWithDetailReport
.\MyTests\Test01_20u.prxres -reportTemplate "Standard Detail Report"
-comment:userCommentTestScenario "First test via outbound proxy server"
```

postfinance_ch_19Dec08_214937_20u.pdf - Adobe Reader

File Edit View Document Tools Window Help

Previous Page Next Page 2 / 51 Previous View Next View Zoom Out Zoom In 125% Scrolling Pages One Full Page Find

Proxy Sniffer - Detail Report

1. Test Scenario

Initial Status	
Test Start Date:	19 Dec 2008 21:49:37
Load Test Program:	postfinance_ch
Load Source Host(s):	fischer (10.8.0.1)
Load Source OS:	Windows XP
Target Host(s):	www.postfinance.ch:80
Applied HTTP Version:	1.1

Comment:
First test via outbound proxy server

Test Input Parameter	
Concurrent User:	20
Planned Test Duration:	unlimited
Planned Loops per User:	2
Startup Delay per User:	200 millisec
Request Timeout per URL:	60 sec
Statistic Sampling Interval:	10 sec
Additional Sampling Rate per Web Page Call:	100 %

11.69 x 8.27 in

5 Manufacturer

Ingenieurbüro David Fischer AG, Switzerland | A company of the Apica Group

Product Web Site: <http://www.proxy-sniffer.com>

All Rights Reserved.