
Slingshot Designer's Guide

Slingshot Version 2

Document Version 1.6
Mar 2007

For more information contact:

Swissrisk AG
Technical Support
Holzhausenstrasse 44
D-60322 Frankfurt
Germany

Phone +49 69 50952 111
Fax: +49 69 50952 199
Email hotline@swissrisk.com
Website www.swissrisk.com

Copyright Information

This document is protected by copyright law and may not be reproduced or distributed either in part or in total. The licensee is not allowed to pass on the software or the accompanying written materials to third parties or make them otherwise available without prior written agreement of the licensor. Information in this document that refers to possible product extensions or to available accessories is not legally binding, especially because the products are subject to continuous adaptation and because the information may also relate to future development. The contents of this document can change without prior notice and does not represent any legal obligation on the part of Swissrisk AG.

Swissrisk AG cannot be made liable for the correctness of information in this document nor for damages resulting from the use of this information or the impossibility of using this information. All other legal regulations for using the software and the corresponding documentation are set in the applicable license agreement.

Slingshot is a trademark of Swissrisk AG. All other product and company names mentioned in this manual are trademarks of their respective companies

Published by:

Copyright © Swissrisk AG | All rights reserved

Swissrisk AG
Holzhausenstrasse 44
D-60322 Frankfurt
Germany

Phone: +49 69 50952-0
Fax: +49 69 50952-333
Website: www.swissrisk.com

Slingshot Designer's Guide

Preface.....	1
What you should know.....	1
General Layout.....	1
InVision Network.....	1
Introduction to Slingshot Publishing.....	2
Features.....	2
Packaging.....	9
Introduction to SLS File Basics.....	10
Introduction.....	10
SLS File Sections.....	10
SLS File Comments.....	10
Inserting the Slingshot Client Viewer into an HTML Document.....	11
Introduction.....	11
Preferred Set-up.....	11
Alternative Set-up.....	12
Browser Differences – Slingshot packages.....	12
Over-riding SLS Parameters in the HTML file.....	12
Choosing a Size for the Slingshot Client Viewer Applet.....	13
Running the Slingshot Client Viewer as an application.....	13
SLS Parameters – HTML Parameters.....	14
Introduction.....	14
Parameters in the [Slingshot] Section.....	14
URL.....	14
Server.....	15
Port.....	15
Feeder.....	15
Quotelist.....	15
Record.....	15
ShowRequestBar.....	15
ShowContributionBar.....	15
ChatApplet.....	16
ObjectStatusMessage.....	16
FontName.....	16
ConnectingMessage.....	16
HeaderLabels.....	16
RowLabels.....	16
HighlightTimeout.....	17
NoChangeImage.....	17
UpImage.....	17
DownImage.....	17
NoLogging.....	17
NoMerge.....	17
AuthorisationID.....	17
ApiDebug.....	18
The LinkToURL and URLTargetWindow Parameter.....	18
LinkToURL in a Quotelist.....	18
LinkToURL in a single Quote.....	18
URLTargetWindow.....	18
Colour Parameters.....	19
NormalColour.....	19
HighlightColour.....	19
MergedHighlightColour.....	19
UpHighlightColour.....	19
DownHighlightColour.....	19
StaleColour.....	20

Tick Colours.....	20
UpTickColour.....	20
DownTickColour.....	20
NoTickColour.....	20
Alternating Colours.....	20
AlternatingBGColours.....	20
AlternatingBorderColours.....	21
AlternatingTextColors.....	21
Parameters in the [Header] Section.....	21
Fields.....	21
Fields Example.....	21
Label and Data Bundles.....	22
Optional SLS File Sections – ConnectingMessageDataBundle.....	23
An Example Quotelist (Figure 1.3).....	24
An Example Quote (Figure 1.2).....	26
An Example Page (Figure 1.4).....	29
Introduction.....	29
Changes to SLS Files.....	31
What's New.....	31
What's Old.....	31
Creating a Slingshot Ticker.....	32
An Example Ticker (Figure 1.9).....	33
Appendix A - Specifying Colour Parameters.....	36
Introduction.....	36
Creating colours.....	36
Textual Colour Markup Scheme.....	36
Hex Colours.....	37
RGB Colours.....	37
Table of Sample Colours.....	37
Appendix B - Specifying Font Parameters.....	38
Introduction.....	38
Up/Down Arrows.....	38
Appendix C – Recommended Maximums per Browser.....	39
What is a Virtual Machine?.....	39
Appendix D - Browser Versions.....	40
Abbreviations :	40
Appendix E - Client Hardware Machine Specifications.....	40
Appendix F - Frequently Asked Questions.....	41
How do I debug the Client Viewer?.....	41
What if the Client Viewer does not load properly?.....	41
What is a “Hard Refresh”?.....	41
How do I do a “Hard Refresh” in Internet Explorer?.....	41
How do I do a “Hard Refresh” in Netscape?.....	41

Preface

This document is intended for "Web Developers" who want to use Slingshot to publish real-time data on the Internet. The live data can be generated internally or supplied by an external feed provider. Swissrisk AG is not a data provider, but we can connect to and publish data from many sources.

Slingshot Client Side requests and formats live data based on parameter settings within configuration files. These files are called SLS files. This document outlines these settings in detail and provides sample SLS files to get you started. Details about how to set-up the Slingshot Web Distribution System (WDS) server can be found in the "Slingshot Administration Guide".

What you should know

This document assumes that you have at least a basic understanding of TCP/IP networks and have a good knowledge of HTML. An understanding of Java Applets and their use is also helpful.

General Layout

In very simple terms Slingshot is based on a client server architecture, whereby a "Client Viewer", usually running in a web browser, requests the data it wants to display from the Slingshot Web Distribution System (WDS). The two machines can be geographically located anywhere, as long as they are both connected to the Internet. They can also, of course, be located on the same machine or an intranet. Figure P1 below shows the Slingshot Client Viewer connecting to the Slingshot WDS that is getting its live real-time data from the InVision network.

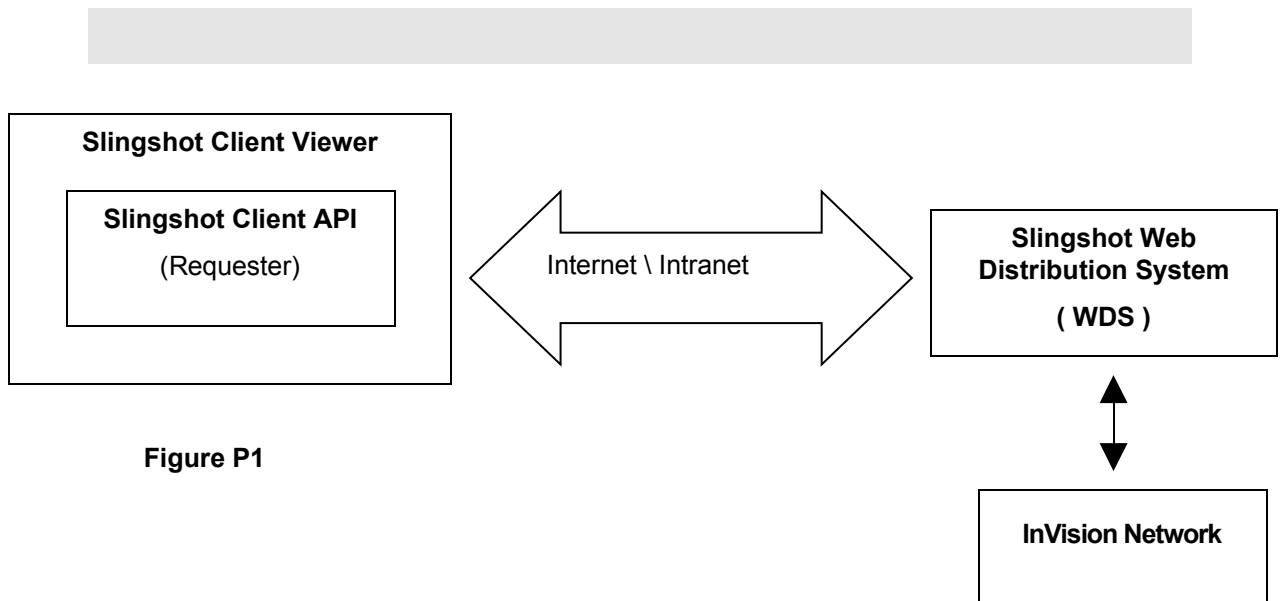


Figure P1

Introduction to Slingshot Publishing

This section provides an overview of real-time data publishing with Slingshot.

Slingshot is a system used to publish real-time data on the Internet. This data can come from any feed vendor or from an internal system. Slingshot is primarily designed for use in presenting Market Data information but can be used for other real-time data delivery applications. It works on both the Internet and on a company intranet.

Slingshot consists of two main components.

- The first component is the Slingshot Web Distribution System (WDS). This is a special server that delivers real-time data to a Slingshot "Client Viewer". This WDS can be run on Windows NT 4.0, Windows 2000, Linux or Solaris. This is documented in the "Slingshot Administration Guide".
- The second component is the Slingshot "Client Viewer". This is a Java Applet that is run in a Java enabled web browser, like Internet Explorer 5 or Netscape 4.76. The Slingshot Client Viewer is responsible for requesting the required data and displaying it on the user's screen. This Client Viewer can also be run as a Java application, removing the need for an Internet Browser. The Slingshot Ticker Viewer is an extension to the Slingshot Client Viewer that scrolls the latest market updates across the screen.

The Slingshot Client Viewer is available in three forms. The first is a simple zip file. It can also be signed, thereby verifying that the application code has not been altered in transit and the Swissrisk AG wrote this code. This is explained in more detail later in this document.

Features

The Slingshot Client Viewer contains the following features:

- **The ability to display quotes**

Figure 1.1 and 1.2 show two methods of viewing the same data. SLS files are used to specify the layout for data and the same SLS file could easily be used to display, in the same format, the data for many different records.

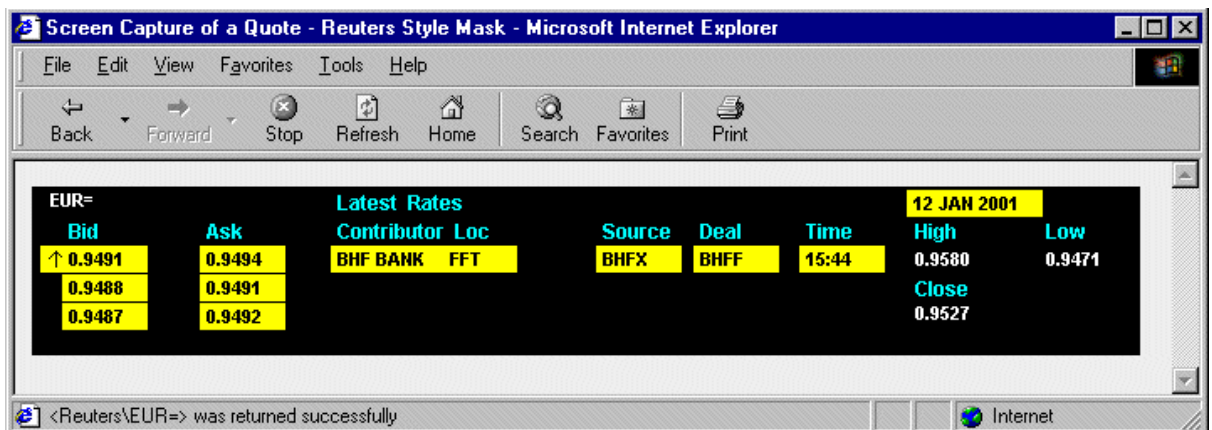


Figure 1.1

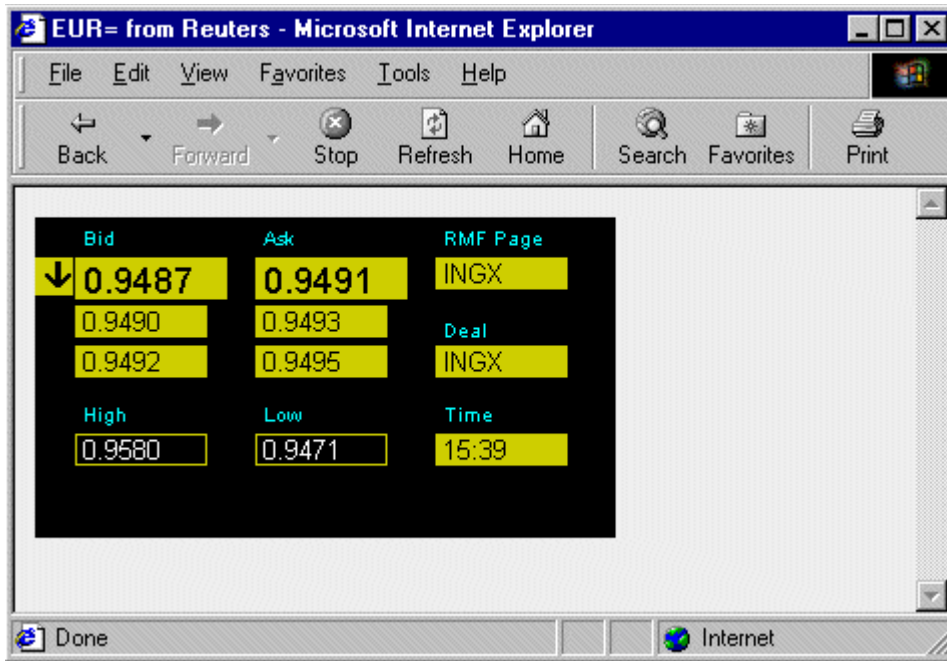


Figure 1.2

- The ability to display quotelists

Figure 1.3 below shows a simple quotelist from Reuters that is formatted similar to that of EFX=.



Figure 1.3

- **The ability to display pages**

The page FFX from Reuters is displayed below in Figure 1.4

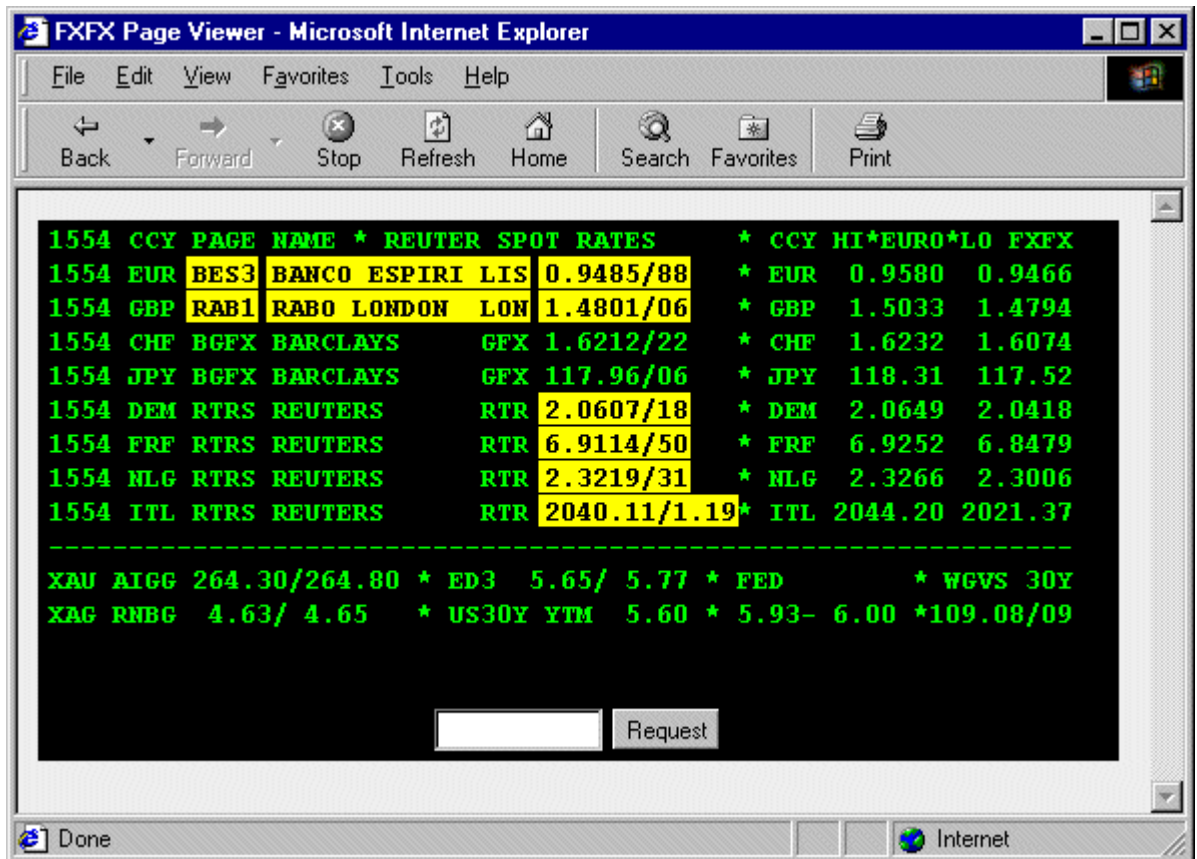


Figure 1.4

- **Request bar**

The above example, Figure 1.4 shows an example of a request bar. The user can enter the name of the record he requires, and press the Request button. The new record will then be requested.

- **LinkToURL**

The Slingshot Client Viewer has the ability to launch web pages at the click of a mouse button. This feature can be configured on a per cell basis in quotes. This feature has powerful dynamic possibilities for use. In conjunction with ASP/JSP or some other dynamic web page creation technologies, news for an item could be displayed, possibly using its current price to compare its performance. Or, a quotelist containing minimal information could link to another frame containing more detailed information about the quote that was clicked.

- The ability to display complex quotelists in a web browser

In Figure 1.5 below VWD's Dax can be seen. It is made up of two separate Client Viewer applets. The top section shows a quote of a single record whereas the lower section is a quotelist.

SYMBOL	ZEIT	NAME	LETZTER	+/-	+/- %	T-HOCH	T TIEF	V-SCHLUSS
846900	16:53:45	DAX (PERFORMANCEINDEX)	6512.72	-47.51	-0.73	6530.57	6480.43	6465.21

F = FORTLAUFENDER HANDEL I = INTRADAY-AUKTION E = Eröffnungsauktion S = Schlussauktion

Symbol	Zeit	Name	Letzter	+/-	+/- %	I.-Preis	GELD	BRIEF	Tagesumsatz
500340	16:48:59	ADIDAS-SALOMON AG	71.73	-0.73	-1.01	72.20	71.99	71.83	346493
840400	16:53:53	ALLIANZ AG O.N.	360.78	-4.08	-1.12	366.99	360.78	360.90	613426
515100	16:53:53	BASF AG O.N.	46.88	-0.35	-0.76	46.02	45.92	45.10	1647391
575200	16:53:39	BAYER AG O.N.	92.18	0.78	1.92	51.70	51.98	52.14	1193393
802200	16:53:37	BAY. HYPO-VEREINSBK	60.94	0.04	0.67	60.16	60.76	60.95	663514
519000	16:53:53	BAY. MOTOREN WERKE	35.38	0.55	1.58	33.21	35.38	35.39	777104
803200	16:53:51	COMMERZBANK AG O.N.	32.14	0.04	0.12	32.40	32.08	32.12	1286197
710000	16:53:45	DAIMLERCHRYSLER AG	43.99	-1.02	-2.27	43.40	43.94	44.04	5743268
542500	16:53:42	DEGUSSA-HUELS AG O	36.50	0	0	35.60	36.37	36.67	163097
514000	16:53:51	DEUTSCHE BANK AG N	93.95	-0.48	-0.42	96.75	93.86	94	3175376
535000	16:53:37	DRESDNER BANK AG N	46.96	0.24	0.51	46.10	46.77	46.96	910269
555750	16:53:49	DT. TELEKOM AG NA	37.53	2.77	7.97	35.69	37.53	37.95	16380246
761440	16:53:39	E.ON AG O.N.	60.07	-1.13	-1.85	61.01	60.03	60.07	2105001
512800	16:53:46	EPCOS AG NA O.N.	99.86	7.53	8.16	98.90	99.90	100	807486
578580	16:53:54	FRESEN. MED. CARE AG	81.90	0.09	0.11	83	81.95	82.10	98865
604843	16:53:38	HENKEL KGAA VZO O.	71	1.04	1.52	67.66	69.45	70	197108
623100	16:53:47	INEINEON TECH. AG N	43.85	2.71	6.63	41	43.88	43.69	3427528
627500	16:53:41	KARSTADT QUELLE AG	35.71	-0.79	-2.16	37	35.71	35.88	415392
648300	16:53:45	LINDE AG O.N.	52.22	-0.73	-1.38	52.90	52.12	52.16	287845
823212	16:53:36	LUFTHANSA AG VNA O	25.52	0.22	0.87	25	25.58	25.52	1169614
593700	16:53:42	MAN AG ST O.N.	29.70	-0.75	-2.46	30.40	29.65	29.79	365043
725750	16:53:28	METRO AG ST O.N.	49.60	-1.40	-2.75	51	49.51	49.68	576995
843002	16:53:51	MUENCH. RUECKVERS. V	347.10	-12.90	-3.58	358.05	347.10	347.85	522655
695200	16:53:44	PREUSSAG AG O.N.	43.65	0.15	0.34	43.98	43.66	43.72	634501
703712	16:53:26	RWE AG ST A O.N.	44.26	-1.13	-2.49	45.39	44.22	44.29	1100071
716463	16:53:49	SAP AG VZO O.N.	179.28	-0.88	-0.44	180	179.28	179.48	1558105
717200	16:53:52	SCHERING AG O.N.	55.29	-1.86	-3.25	57.89	55.10	55.45	863580
723610	16:53:51	SIEMENS AG NA	145.70	3.80	2.68	142.85	145.65	146	2838360
750000	16:48:56	THYSSENKRUPP AG O.	18	-0.12	-0.66	18.19	17.96	18.03	919991
766400	16:53:47	VOLKSWAGEN AG ST O	54.68	0.55	1.02	54.54	54.68	54.84	1821844

[Back to Slingshot.net](#)

Figure 1.5

- **The ability to display a Ticker**

A ticker is available in Slingshot. This ticker scrolls real-time information from right to left on screen. An example can be seen in Figure 1.6 below. In this example it is configured to blend into the upper orange frame in the page.

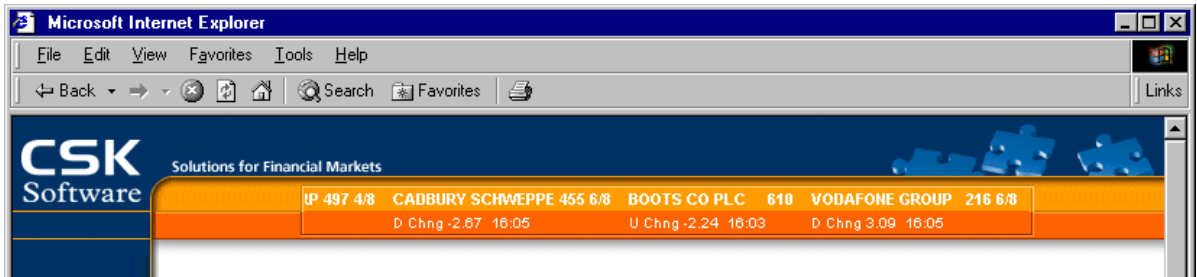


Figure 1.6

- **Merged Highlight colours**

The purpose of a merged update is to prevent there from being too much data on the Network/Internet. The user can see when an update was merged because a different highlight colour will be used.

- **Multiple Client Viewer applets on the one page**

The FTSE 100 is represented on the web page below in Figure 1.7. This consists of one Client Viewer applet at the top, which uses one SLS file, and five similar Client Viewer applets below that which use the same SLS file. The Client Viewer applet is told what records to request by parameters that are passed via the HTML file.

The screenshot shows a Microsoft Internet Explorer browser window displaying an FTSE 100 quotelist. The browser title is 'FTSE quotelist in multiple Applets - Microsoft Internet Explorer'. The page content includes a table with the following columns: LAST, NETCHNG, NETCHNG%, TRDTM, OPEN, CLS, CLS DATE, YRLOW, and YRHIGH. The table contains 100 rows of data, each representing a stock. The first row is highlighted in red, indicating a merged update. The table is organized into a grid of five columns and 20 rows.

LAST	NETCHNG	NETCHNG%	TRDTM	OPEN	CLS	CLS DATE	YRLOW	YRHIGH
6165.5	50.6	0.83	16:37	6114.9	6114.9	11 JAN 2001	6029.3	6276.2
AAL.L 3825	-46			EMLL 571	4 4/8			
ABF.L 493 4/8	-13			EXLL 1049	3			
AL.L 720	-4 4/8			GCP.L 702 4/8	20			
ALLD.L 437	15 4/8			GKN.L 765	2 4/8			
ANL.L 1225	-11			GME.L 415	-4			
ARML 550 4/8	31 4/8			GSK.L 1774	-23			
AUTNL 1984	122			GUS.L 482 4/8	20 4/8			
AVZ.L 1395	-8			HAS.L 359 4/8	5			
AZNL 3099	2			HFX.L 673 4/8	1			
BAL 267 2/8	2 2/8			HGL 224 2/8	1 6/8			
BAA.L 592 4/8	0 4/8			HNS.L 453 4/8	-0 6/8			
BARC.L 2083	-25			HSBAL 1004	13			
BASS.L 784	2 4/8			ICLL 511	-8			
BATS.L 494 4/8	-16			ILL 1303	11			
BAY.L 436 6/8	11 2/8			IMT.L 691 4/8	-8 4/8			
BCIL 488 6/8	1 2/8			IPR.L 224 4/8	-5			
BGL 255	-6 2/8			ISYS.L 173 2/8	0 4/8			
BLT.L 265 6/8	1 2/8			KGF.L 486 4/8	-7 4/8			
BOC.L 1002	-36			LAND.L 859	-36			
BOOT.L 573 4/8	-34			LAT.L 137 4/8	-2			
BP.L 556	9 4/8			LGEN.L 179 6/8	-3 4/8			
BSCT.L 767 4/8	21 4/8			LLOY.L 728 4/8	-4			
BSYL 1199	-10			LOG.L 1598	105			
BT.L 710 4/8	4			MKS.L 200 4/8	2 6/8			
CBRY.L 442 4/8	12 4/8			MONLL 713 4/8	35 4/8			
CCH.L 1035	1			MSY.L 627	39			
CCML 567 4/8	4 4/8			NAM.L 522 4/8	23 4/8			
CGNLL 1000	-24			NGGL 610 4/8	-4			
CMGL 850	101			OML.L 158	-0 6/8			
CNAL 237 6/8	-11 2/8			PRJL 1000	-26			
CP.L 495	12			PSO.L 1615	40			
CTM.L 1767	48			PWG.L 631 4/8	-8			
CWL 971 4/8	14			RB.L 828	-22 4/8			
CWGL 474 2/8	1 4/8			RBOS.L 1610	3			
DDT.L 433	34 4/8			REED.L 637 4/8	1 4/8			
DGE.L 666 4/8	14 4/8			RIOL 1179	15			
DMGOa 887	5			RR.L 201 6/8	0 4/8			
DXNS.L 268 4/8	-5 6/8			RTL.L 995	2			
ECML 677 4/8	-4			RTOL 209 2/8	0			
EGS.L 571	42			RTRL 1091	63			
				RSAL 506 4/8	-11 4/8			
				SAB.L 491 2/8	19			
				SBRY.L 355 6/8	7			
				SDRL 1312	-10			
				SFWL 299	-6			
				SGEL 341 4/8	19 6/8			
				SHEL.L 549 4/8	11			
				SHP.L 1150	41			
				SMIN.L 706	-22 4/8			
				SPT.L 558 4/8	17			
				SPWL 466 4/8	-10 6/8			
				SSEL 571 4/8	-29 4/8			
				STAN.L 1030	18			
				TSCOL 258	-4 4/8			
				TWTL 136 4/8	4 2/8			
				UBML 793 4/8	-12 4/8			
				ULVRL 516	9 4/8			
				UUL 588	-21			
				YODL 236 2/8	10			
				WPP.L 835 4/8	27 4/8			

Figure 1.7

- **Param tags can be over-ridden from the HTML file**

In the quotelists in the above example, Figure 1.7, only one SLS file is used to specify how the quotelists look. The parameters about which records to request are passed via the HTML file, reducing the number of files needed and making administration much easier. See the section "Over-riding SLS Parameters in the HTML file" for more information.

- **Configurable fonts**

All fonts are configurable as can be seen in the above example, Figure 1.7. Here we have multiple fonts. The font used in the quotelists is size 12 "SansSerif" in bold. The font which is used for the up arrow is "ZapfDingbats" and the larger font used in the upper Client Viewer is size 17 "SansSerif".

- **Automatic Quotelist Sizing**

The heights of Cells in a quotelist are automatically set in accordance with the size of the font used. This makes configuration easier.

- **Removal of header labels**

Header Labels can be removed as show in the above example, Figure 1.7. This saves space when it is understood that the user knows that the current price and the daily change value are being displayed.

- **Removal of row labels**

Row labels can also be removed from quotelists if you wish. This means that the Slingshot Client Viewer will not display the name of the record being requested. Figure 1.5 shows an example of a Quotelist with the automatic row label removed.

- **Dead/Live colours**

The user can see when the data is stale, i.e. not up to date, because the data on their screen changes to a configurable colour combination. The example below, Figure 1.8, shows red text on a black background as being the "Stale" colour combination.

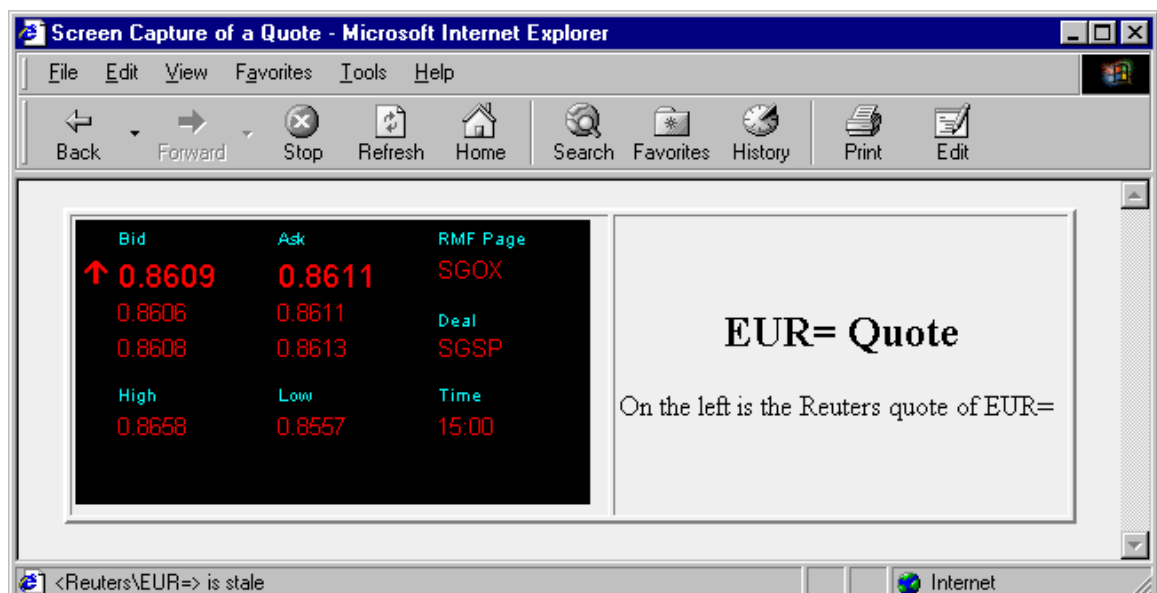


Figure 1.8

- **Configurable Highlight timeouts**

It is possible to set how long an object is highlighted for. This can vary between 10 milliseconds and about 50 days!

- **Colour input in Hex, RGB or text formats for ease of use**

Example of these can be seen in the example quote, quotelist and page in the sections following. More details about how to specify colours can be seen in Appendix A.

- **Invisible Cells are supported**

It is possible to include a cell in a single quote without it being visible on the screen.

- **Scroll bars for quotelists**

If according to the size of the Client Viewer applet a quotelist needs a scrollbar it will automatically be added. This is very useful for very long quotelists like Reuters' FX=.

- **Configurable connection messages**

It is possible for you to change the initial connection message in the Slingshot Client Viewer.

- **Images for up/down Ticks**

This can be seen in the example in Figure 1.3. The images showing up/down ticks allow the user to easily see the progress of the record they are watching. This is source dependent.

- **Small download size**

The Slingshot Client Viewer size varies between 62 kilobytes and 97 kilobytes depending on how it is packaged. This is a significant improvement over Slingshot 1, which was greater than 300 kilobytes.

- **Efficient**

Slingshot does not send data that is not requested. Only the fields that are required are sent to the user. This dramatically reduces the bandwidth that Slingshot Version 2 uses over Slingshot Version 1.

- **Just one Requester per Web Page**

The ClientAPI that runs under the Slingshot Client Viewer ensures that only one connection is made to the Slingshot WDS per web page. This reduces the bandwidth requirements and makes the system more efficient. This is also an improvement over Slingshot 1, which had multiple connections per web page.

- **Merged updates**

Also, the default behaviour for the ClientAPI is to limit the bandwidth used. There is a maximum amount of data sent per client per second. When the clients connection exceeds this specification updates are merged. The client will still get the most up to date data and they will be informed, via the colour of the highlight, that some intermediate data was not sent.

- **Secure**

Slingshot can be configured to use Secure Sockets Layer encryption. This is the Internet industry standard that is used to ensure that data is not interfered with en-route to you. It provides protection against eavesdropping, tampering and forgery.

- **Status messages**

Status messages inform the user of the progress of the applet, and of problems when they arise. A status message, "<Reuters\EUR=> is stale", can be seen in the status bar of the browser in Figure 1.7

- **Delayed Data**

If not permitted for a real-time data object the WDS can be set up to send delayed data. This data is displayed in the Viewer with a red dashed strike-through.

- **Statistics**

It is possible to view some statistics about the running of the Slingshot Applets. To view these statistics select either the Standard Viewer or the Ticker Viewer with your mouse and press the F10 key. Your Java Console will now contain the latest information gathered about the current applets running. See "Appendix F – Frequently Asked Questions" for information on how to view debugging information in the Java Console. These statistics include the number of SingleHttpConnection classes (this basically means the number of objects being watched) and the length of time that the Applets have been running. Information about the number of updates received and the average number of updates received per second is also displayed. This is displayed both since the applet was started, and since you last viewed the statistics.

- **Contributions**

It is possible to contribute data using the Slingshot Client Viewer, via the WDS to a contribution service, for example, an RDS. This is possible for both single quotes and quotelists.

- **Chat Applet**

Using the contribution powers of the Client API it is possible to configure an RDS, WDS and applet to create a chat program. Users simply contribute to a predefined FID in the WDS and all users watching that FID, usually with the Ticker Viewer, can read the information. This feature could be used in an organisation either formally or informally, with a variety of different "threads" in place at the one time.

Packaging

The Slingshot Client Viewer comes packaged in three ways. All three have the title SLSViewer.TYPE, where TYPE is one of the three types mentioned below.

The simplest package is called a ZIP file. This file can be used by any Java enabled Internet browser. Its purpose is simply to compress the Java files, reducing the download time for Slingshot.

The second package is called a CAB file. This file is signed and can only be used by Internet Explorer. The reason for signing this file is to prove to your Clients that this Java archive came from Swissrisk AG and can therefore be trusted. On download of the Slingshot Client Viewer applet Thawte Consulting (Pty) Ltd. (www.thawte.com) automatically confirms whether or not Swissrisk AG developed this exact, unchanged applet. Files in this format are also compressed keeping the download size to a minimum.

The third package format is called a JAR file. Netscape uses this in a similar way to the CAB file is used by Internet Explorer. Once again, on download of the signed Slingshot Client Viewer applet Thawte Consulting (Pty) Ltd. (www.thawte.com) confirms whether or not this exact, unchanged applet was developed by Swissrisk AG. Files in this format are also compressed keeping the download size to a minimum.

Note: See "Inserting the Slingshot Client Viewer into an HTML Document" in this document for details on how to use the different packages in your web page.

Introduction to SLS File Basics

Introduction

An SLS file is a plain text file that is edited by the web developer/designer. It contains the details about what real-time data to view and how it should look within the Slingshot Client Viewer. The positioning and size of this applet is controlled by the HTML file and is discussed later in this document.

Depending on the server configuration the SLS file is stored on the same machine as the WDS or else the same machine as the HTML Web Server. The details of possible server configurations can be found in the "Slingshot Administration Guide" Documentation.

SLS File Sections

The SLS file is broken up into two main sections, which in turn define more sections. A section is a word surrounded by square brackets.

Example

```
[Slingshot]
```

The two sections required in an SLS file are `[Slingshot]` and `[Header]`. The contents of these two sections are described in the next section.

SLS File Comments

SLS files can contain comment lines. These lines are for the designer only and are ignored by the Client Viewer that is parsing the file. It is good practice for a designer to comment any SLS files he writes or changes. Comments lines must begin with a hash (#) character or a semi-colon (;) character.

Example

```
# This SLS File displays real-time Reuters information  
; about the main European currencies in the form of  
# a quotelist. Written by Richard Mangan.
```

Inserting the Slingshot Client Viewer into an HTML Document

Introduction

This section explains how the Slingshot Client Viewer is embedded in a web page for viewing on the Internet. It also explains how to over-ride SLS file parameters in your HTML file. There are two likely set ups which will be explained briefly. These are explained in more detail in the "Slingshot Administration Guide" documentation. The main difference between these two set-ups is the `CODEBASE` parameter.

Preferred Set-up

In the preferred set-up two machines are used on the Server Side.

The first machine is a web server. This stores and serves all the HTML files for the organisations web site. It should use a standard HTTP web server like Apache (www.apache.org).

The second machine will be used solely for the Slingshot WDS. This program not only serves real-time data to the appropriate client but it also serves the HTTP requests for the packaged Slingshot Client Viewer and the SLS files. In this set-up the SLS files are stored on the Slingshot WDS machine whereas the HTML files are stored on the web server machine.

Example

```
<APPLET CODEBASE="http://Slingshot2Server:80" ARCHIVE="SLSViewer.zip"
CODE="net.Slingshot.Viewer.Viewer.class" WIDTH=400 HEIGHT=200>
<PARAM NAME="CABBASE" VALUE="SLSViewer.cab">
<PARAM NAME="UniqueName" VALUE="myFirstHTMLFile">
<PARAM NAME="SRC" VALUE="sls/Master.sls">
Slingshot Server dead or Java not enabled.
</APPLET>
```

Note: The sentence "Slingshot Server dead or Java not enabled" is needed so that the user knows when the WDS is dead. When the WDS is dead the applet cannot normally be served and therefore the applet cannot show a status message. This message is displayed by the browser instead.

Note: The full path from the root of the "Slingshot2Server:80" must be given for SLS files. Paths relative to the html file are not accepted.

Alternative Set-up

The alternative set-up involves just one machine. This will mean that this machine will serve both the web site and the real-time data. However there are two main issues to be aware of:

1. This is only suitable for small installations. Due to the extra load on the Server machine from having to serve live real-time data only installations expecting to receive few concurrent logins, or which have only a small number of hits on their web site should use this solution.
2. Another problem is that there is less chance of HTTP tunnelling being successful because the Slingshot WDS will be publishing on a port other than the standard HTTP port, 80.

Example

```
<APPLET CODE="net.Slingshot.Viewer.Viewer.class"
ARCHIVE="SLSViewer.zip" WIDTH=400 HEIGHT=200>
<PARAM NAME="CABBASE" VALUE="SLSViewer.cab">
<PARAM NAME="UniqueName" VALUE="myFirstHTMLFile">
<PARAM NAME="SRC" VALUE="sls/Master.sls">
Slingshot Server dead or Java not enabled.
</APPLET>
```

The parameter **UniqueName** should always be added to the applet tag. The use of UniqueName is important because it allows the internal Client API, the requester, to know which instance of the Slingshot Client Viewer has requested which records and this means that only one connection is made to the Slingshot WDS. This is important in reducing the bandwidth used.

Browser Differences – Slingshot packages

In the two examples above the packages used are the ZIP file and the CAB file. Internet Explorer, not the Slingshot Client Viewer, reads the CABBASE applet parameter and if present it attempts to download the signed CAB archive.

If you do not want to use signed applets you simply remove this CABBASE applet parameter. The default ZIP file, which is unsigned, will be used instead. This ZIP file is compatible with all browsers.

Over-riding SLS Parameters in the HTML file

SLS file parameters can be over-ridden for a few reasons. The first reason is to ensure that only a minimum number of SLS files need to be written. So in a simple environment only two SLS files would be needed. One would be for quotes and the other would be for a quotelist. The quotelist could display different data according to the Quotelist parameter as set in an HTML file.

The second, equally important reason is to allow the web designer produce a more interactive experience. The <param> tag could be filled out with some JavaScript or perhaps an ASP page. A user could therefore select from a list what Records he would like to see in his personal quotelist from a JavaScript list box.

How to over-ride parameters

Within the <APPLET> </APPLET> section in the HTML file add the following to over-ride, or add, a parameter in the SLS file:

```
<param Name = "ParameterName" Value = "Parameter Value">
```

The first example below changes the Highlight timeout parameter to be one second. The second example sets the quote being used to be EUR= on the service ReutersSFPLUS

Example 1

```
<param Name="HighlightTimeout" Value="1000">
```

Example 2

```
<param Name="Feeder" Value="ReutersSFPLUS">
```

```
<param Name="Record" Value="EUR=">
```

Choosing a Size for the Slingshot Client Viewer Applet

When viewing a quotelist with the Slingshot Client Viewer the applet will automatically add scrollbars as required. If you do not want to see scrollbars and want to make sure that the applet is sized exactly the height of the applet can be calculated as (the number of rows) * 15. The width is as you specified while designing the SLS file.

Note that the Slingshot Client Viewer does not add scrollbars when viewing single quotes.

Running the Slingshot Client Viewer as an application

The Slingshot Client Viewer can be run as a Java application if required. When running as a Java application the Client Viewer uses much less memory because a browser does not need to be loaded into memory.

The Client Viewer needs the following present to run as an application

1. A Java Virtual Machine, minimum version 1.1, must be present on the machine. A Java Virtual Machine can be downloaded from <http://java.sun.com/j2se/1.3/>
2. The SLSViewer.zip file must be present.
3. At least one SLS file must be present.

A batch file should be created to run the Client Viewer as an application. All parameters in the following example can be changed unless otherwise stated.

```
java.exe -classpath SLSViewer.zip net.Slingshot.Viewer.Viewer %1
```

java.exe – the name of the Virtual Machine. A common variation is jview.exe

-classpath – This allows a class path to be added. A common variation is -cp

SLSViewer.zip – This is the name of the compressed file containing the Viewer executable files. This can not be changed.

net.Slingshot.Viewer.Viewer – This is the name of the executable file

%1 – This should be replaced by the name of the SLS file to be requested.

SLS Parameters – HTML Parameters

Introduction

Parameters are used to specify how something looks in the Slingshot Client Viewer. They consist of the parameter name followed by the equals (=) character followed by the parameter's value. This value may be an integer (a positive number), a textual value (a word or a sentence) or a textual representation of true or false (true/yes or false/no). Details of the type required are specified with each parameter. SLS files are case **insensitive** which means that the word "sLiNgShOt" is equivalent to "Slingshot". All the parameters in this section are used in the same way when creating both quotelists or quotes unless stated otherwise.

Parameters in the [Slingshot] Section

All the parameters in the [Slingshot] section can be over-ridden in the HTML file. How to do this is described in the section "Over-riding SLS Parameters in the HTML file". The reason for this functionality is so that a standard SLS file can be used for, for example, a quotelist, and the actual contents of the quotelist can simply be changed in the HTML file, perhaps using JavaScript or ASP technologies.

URL

This parameter specifies where the Slingshot requester, internal to the Slingshot Client Viewer, should request the real-time data from.

Example

```
URL=http://127.0.0.1:80/Reuters/IEP=
```

This parameter is of type URL and is broken up as follows.

It begins with http://

This specifies the protocol to be used. The Secure Sockets Layer (SSL) protocol can be used if your browser supports it and the WDS server is configured to supply secure data.

It is followed by the IP Address of the Slingshot WDS. This can be the host name of the machine but the IP Address is always recommended. Please note that this must be the external IP Address of a machine that is on the Internet.

This is followed by the name of the Service/Feeder where you would like to request the data from. The Slingshot WDS can deliver data from multiple data services and this part of the parameter defines which one is required.

This is followed by the name of the record to be requested. If it is a single quote a single record name appears here. If it is a quotelist then the list of record names should be separated by commas.

Server

This parameter defines the protocol and machine to connect to. This parameter over-rides any Server value set in the URL parameter.

Example

```
Server=http://127.0.0.1:80
```

Port

This parameter is used to define the port on the Slingshot WDS machine to connect to. The default is 8000 but it is often set to 80. This parameter over-rides any Port value set in the URL parameter.

Example

```
Port=80
```

Feeder

This parameter specifies the Service to connect to on the Slingshot WDS machine. This parameter over-rides any Service/Feeder value set in the URL parameter.

Example

```
Feeder=ReutersSFPLUS
```

Quotelist

This parameter specifies the list of objects that are to be requested from the Slingshot WDS. This parameter over-rides any values set in the URL parameter. It also over-rides the Record parameter below.

Example

```
Quotelist=IEP=,EUR=,ESP=,ITL=,DEM=,FRF=,JPY=
```

Record

This parameter specifies which record is to be viewed in a quote. This parameter over-rides any values set in the URL parameter but if the quotelist parameter is set it is over-ridden.

Example

```
Record=EUR=
```

ShowRequestBar

This parameter allows a request bar to be added to the bottom of the Client Viewer. It only works with single quotes, not with quotelists. The default is FALSE or NO. It is possible to request an object from a different service if the separator ` is used.

Example

```
ShowRequestBar=True
```

ShowContributionBar

This parameter allows a special contribution bar to be added to the bottom of the Client Viewer. It works with both single quotes and quotelists. The default is FALSE or NO. To Contribute to an object ensure that it has been requested.

1. Double click on the FID value which you want to update. The current value of that FID will appear in the contribution text box.
2. Select the contribution text box and edit the current value to the new value.
3. Press "Enter" or else press the "Add To List" button. The list box below will now contain the FID number for that FID, and the new value.

4. Continue step 1 to 3 until you have entered all the changes you wish to make. You may delete entries by highlighting them in the list box and pressing the "Delete" button.
5. Press the "Contribute" button to send the data to the WDS. Information regarding the contribution will be displayed in the Java Console. If the contribution is successful the Object will update with your new values.

Example

```
ShowContributionBar = True
```

ChatApplet

If ShowContributionBar has been set to TRUE it is possible to set the contribution bar to be styled for use in a chat applet. The user is also prompted for a user name when the applet starts. They must enter a name. The user can then enter what they want to say and press either the Enter key or the "Chat" button.

Example

```
ChatApplet = True
```

ObjectStatusMessage

This parameter specifies whether or not status messages should be displayed in the browser's status bar as well as the Java Console. The default is FALSE or NO.

Example

```
ObjectStatusMessage=Yes
```

FontName

This specifies the default font for the Client Viewer. In **quotelists** it is also the font that is used by the column containing the Record Names, if applicable (See the RowLabels parameter below). The default font is plain Courier of size 11. See Appendix B for more details about specifying fonts.

Example

```
FontName = ~TimesRoman~ 10 BOLD
```

ConnectingMessage

This is the message that is first displayed when the Client Viewer is finished loading. Once requests are made then these new messages appear in the applet. The default message is "Connecting to Slingshot Server...". See the section "Optional SLS File Sections – ConnectingMessageDataBundle" for more details.

Example

```
ConnectingMessage = Connecting to Slingshot...
```

HeaderLabels

This parameter states whether or not to have labels at the top of a **quotelist**. If switched off no labels are displayed at the head of the columns. The default is TRUE or YES.

Example

```
HeaderLabels=FALSE
```

RowLabels

This parameter states whether or not to have labels on the left-hand side of each row (i.e. the Record Name) of a **quotelist**. If switched off no labels are displayed for each row. The default is TRUE or YES.

Example

```
RowLabels=FALSE
```

HighlightTimeout

The amount of time that a cell remains in the highlight colours after it has been updated is specified by this parameter. It is measured in milliseconds, not in seconds. The default value is 3000 milliseconds, which is three seconds.

Example

```
HighlightTimeout=1000
```

NoChangeImage

This specifies the image name to be used when either FID 115 or 14 is not changed and the images parameter is set to true. The default value is "same2-15.gif".

Example

```
NoChangeImage=noChangeSize16.gif
```

UpImage

This specifies the image name to be used when either FID 115 or 14 is higher than the previous value and the images parameter is set to true. The default value is "up2-15.gif".

Example

```
UpImage=upImageSize16.gif
```

DownImage

This specifies the image name to be used when either FID 115 or 14 is lower than the previous value and the images parameter is set to true. The default value is "down2-15.gif".

Example

```
DownImage=downImageSize16.gif
```

NoLogging

This parameter allows the designer to switch off any console logging which might be considered a security risk like IP Addresses or host names. It makes debugging more difficult and the information is available in the SLS file anyway so its default is FALSE.

Example

```
NoLogging = TRUE
```

NoMerge

When set this parameter informs the WDS that updates should not be merged before sending them. This ensures that all updates are received in the Client, although the Server's HTTP_ASSEMBLE_TIME still applies (see the Administration Guide for more details). The default is FALSE.

Example

```
NoMerge = TRUE
```

AuthorisationID

This parameter allows the passing of an Authorisation ID to the WDS. See the Administration Guide for more details. The default is an empty string.

Example

```
AuthorisationID =MyPassword
```

ApiDebug

This parameter can be set to true to log client API debug information to JAVA console. Default is false.

The LinkToURL and URLTargetWindow Parameter

Slingshot provides the ability to link to other web pages, from the Client Viewer, as required. This linked page can be anything from an internally generated ASP page using the current information in the cell to an externally available web site. The user can tell when a cell is a LinkToURL cell because the cursor will change to a hand shaped pointer.

Note: These parameters are only valid when the Slingshot Client Viewer is run as an applet, and **not** when it is run as a Java application.

The LinkToURL parameter takes the form of an URL, with two possible extensions. It is possible to pass the name of the Object that the cell refers to, and/or the current data in the cell. These features are particularly powerful when used in an ASP environment.

The optional extensions are "%n" and "%s".

Command	Replaced By	Example
%n	Name of the Object for which this cell was created	http://127.0.0.1/News/%n.jsp
%s	The current text in the cell	http://127.0.0.1/Search/%s.asp

Note: Full URL's must be stated including the "http://" section.

LinkToURL in a Quotelist

The LinkToURL parameter is specified in the [Slingshot] section when used in a quotelist. All cells in the quotelist will use the same link string, possible going to different pages due to the "%s" and "%n" optional extensions.

LinkToURL in a single Quote

The LinkToURL parameter is specified in the [Slingshot] section if you want all cells in a quote to perform the linking. In this case all data cells in the quote will use the same link string, possibly going to different pages due to the "%s" and "%n" optional extensions.

If you only want specific cells to perform the function you should place the parameter in the required section, ensuring that you do not have a LinkToURL parameter in the [Slingshot] section. In this case only the specific data cells in the quote will use the link string. Different cells can then point to different web pages.

URLTargetWindow

This parameter should be placed in the same place as the LinkToURL parameter. It does not have any function if the LinkToURL parameter is not present. If this parameter is not specified the page from the applet is replaced with the new page. This parameter specifies which browser frame the link should appear in. Otherwise the frame with the specified name is opened. If no frame with the specified name is currently available a new browser frame will be created.

Example

```
URLTargetWindow=bottomFrame
```

Colour Parameters

More information about colour parameters is available in Appendix A. It is recommended that you read that section before proceeding.

NormalColour

This is the standard colour for the Client Viewer. The Client Viewer is displayed in these colours when the service is live and no update has been received within the HighlightTimeout period. It takes up to a maximum of three colours. The default colours are green text on a black background with a black border.

Example

```
NormalColourText = black on white on black
```

HighlightColour

The highlight colour is used in conjunction with the HighlightTimeout parameter specified above. When an update is received the colouring of the specific cell is changed so that it is easily visible to the user. If you simply want the border to show that the colour has updated try the following in conjunction with the NormalColour parameter above. The default colours are black text on a green background with a black border.

Example

```
HighlightColourText= black on white on blue
```

MergedHighlightColour

The merged highlight colour is used in conjunction with the HighlightTimeout parameter specified above. A merged update is one where the latest update is sent to the client by the server, but intermediate updates have been skipped due to limitations of the Internet connection. This means that the user is seeing the latest data but they are being informed that some of the updates prior to this update have been missed. Please see the "Slingshot Administration Guide" for more details. The default colours are black text on a cyan background with a black border.

Example

```
MergedHighlightColourText= black on lightgrey on lightgrey
```

UpHighlightColour

The up highlight colour is used to inform the user, when an update occurs, that the numerical value in the cell has increased. If you want the user to be permanently informed of the status of the latest update then the "UpTickColour" parameter should be used. The UpTickColour parameter sets the colour of the Cell's text, after the highlight has completed. These two parameters can be used at the same time, if you wish. The "UpHighlightColour" parameter has no default colours as it is only activated if required.

Example

```
UpHighlightColourText= black on lightgreen on lightgreen
```

DownHighlightColour

The down highlight colour is used to inform the user, when the update occurs, that the numerical value in the cell has decreased. If you want the user to be permanently informed of the status of the latest update then the "DownTickColour" parameter should be used. The DownTickColour parameter sets the colour of the Cell's text, after the highlight has completed. These two parameters can be used at the same time, if you wish. The "DownHighlightColour" parameter has no default colours as it is only activated if required.

Example

```
DownHighlightColourText= black on lightred on lightred
```

StaleColour

This is the colour that the Cells in the Client Viewer turn to when they are marked as stale. Stale means that the service where the data was sourced (e.g. Reuters or VWD) has either gone dead or is no longer supplying live real-time data. There are numerous reasons why this could happen. The reason for a stale colour is so that the user knows that the data they see on the screen is not recent. The default stale colour is red text on a black background with a black border.

Example

```
StaleColourText = lightred on black on black
```

Tick Colours

These optional parameters allow you to specify colours to show whether a numerical value has increased or decreased. A calculation is made against the previous value of the cell and it is coloured, after highlighting, accordingly. If the calculation fails, as in a textual field, the original colour settings remain. They have no defaults and are unused if not specified. These colours are visible only after the highlight has disappeared. If you want the user to know, on highlight, the status of the update use the "UpHighlightColour" and "DownHighlightColour" parameters mentioned above. These parameters are not feed dependent. They rely on the previous numerical value in a cell to decide the colour.

UpTickColour

This is the colour that is used when the cell can be calculated as having increased.

Example

```
UpTickColourText=lightgreen
```

DownTickColour

This is the colour that is used when the cell can be calculated as having decreased.

Example

```
DownTickColourText=lightred
```

NoTickColour

This is the colour that is used when the cell can be calculated as having remained unchanged.

Example

```
NoTickColourText=blue
```

Alternating Colours

Alternating colours are used only in quotelists to make rows of data more visible. Unlike other colour settings these are lists of up to 10 colours, all for the same feature. Alternating colours over-ride NormalColour settings and column colour settings. The colours repeat if required.

AlternatingBGColours

The list of background colours to use. The example uses two alternating background colours.

Example

```
AlternatingBGColoursText = white lightgrey
```

AlternatingBorderColours

The list of border colours to use. The example uses two alternating border colours.

Example

```
AlternatingBorderColoursHex = #FFFFFF #C0C0C0
```

AlternatingTextColors

The list of text colours to use. The example uses three alternating text colours.

Example

```
AlternatingTextColorsRGB = 0,255,0 0,192,0 64,255,64
```

Parameters in the [Header] Section

There is just the one parameter in the [Header] section. This is specified below. This parameter can be added to the [Slingshot] section to avoid the need for a [Header] section. Either choice is acceptable.

Fields

This parameter is the starting point for specifying how the data is displayed in the Slingshot Client Viewer. Examples of how to create a quotelist, a quote and a page are available below.

Example

```
Fields = Tick Bid Ask Time
```

The Fields line contains the names of the user-defined sections where information about the layout and appearance are stored. These sections vary according to whether you want to view a quotelist or a quote. There are sections giving examples of a quotelists, a single quote and a page.

In general the User Defined section from the fields line contains the following pieces of information.

Fields Example

```
[Tick]
```

```
LabelPosition = 10 15
```

LabelPosition is mandatory in quotelists. Only the first value is used, i.e. the x co-ordinate of the column measured in pixels. In single quotes the LabelPosition can be used to position a non-updating piece of information on the screen. CellWidth can be used in this case.

Note: LabelPosition is not required in Slingshot 1 quotelists where a DataFormat line is used. This is for backward compatibility only and should be avoided. See Appendix C for details of Slingshot 1 features not supported.

```
Label=Bid
```

The Label parameter must exist in a quotelist. It can however be blank if, for example, no label is required. The Label parameter is not required for quotes.

```
LabelBundle = BidLabelBundle
```

The LabelBundle parameter specifies the user-defined section where details about the appearance of the Label can be found. If a label exists this section must exist or else no label will be shown.

DataPosition = 10 35

The DataPosition parameter is used in quotes only. Quotelists position the Data according to the LabelPosition parameter above. It must take two numbers, an x co-ordinate followed by a y co-ordinate.

DataSource = 115

The DataSource parameter specifies the Field ID of the data being requested. Each piece of data in a financial record is numbered. This number is required for requesting specific fields from a Service. These values can not be alphabetic characters.

DataFormat = %-3.3s

This is for backward compatibility purposes only. It used to be used to layout quotelists but more precise pixel widths are used instead.

DataBundle = BidDataBundle

The DataBundle parameter is the same as the LabelBundle parameter above. It is used to define the appearance of the Data in both quotelists and quotes.

Label and Data Bundles

Label and data bundles are used for the same purpose, to define how a component looks.

Example

[TickDataBundle]

FontName = ~TimesRoman~ 12 BOLD

The FontName parameter is used to specify which font to use. See Appendix B for more details about this parameter.

ColourText = white on blue on black

The Colour parameter specifies the colours to be used by any cells that have this section as their Data/LabelBundle. See Appendix A for more details about colours.

Images = FALSE

The Images parameter allows you to add images to Data Sources or either 14 or 115. See Appendix B for more details about this.

CellWidth = 50

This can be used to set the width of a cell. It is always applicable for single quotes and pages but is only used in the right-most column in a quotelist. In this case, it must be put in the LabelBundle section of the SLS file.

LabelRightAlign = TRUE

It is possible to align the label to the right hand side of a Cell if you wish. The default is FALSE which aligns it to the left.

DataRightAlign = TRUE

It is possible to align the data to the right hand side of a Cell if you wish. The default is FALSE which aligns it to the left.

Optional SLS File Sections – ConnectingMessageDataBundle

There is an optional SLS file section that is called ConnectingMessageDataBundle. This section contains details about how the connecting messages look. These messages inform the user of the progress of the requests to the Slingshot WDS and the current state of the requester if there is a problem. The default for this is the default Slingshot font in red on a black background. See the Appendices for details.

Example

```
[ConnectingMessageDataBundle]
FontName = ~Courier~ 16 BOLD
ColourText = lightBlue on Black
```

An Example Quotelist (Figure 1.3)

```
[Slingshot]
URL = http://127.0.0.1:80/Reuters/ignore
Quotelist=EUR=,JPY=,GBP=,CHF=,DEM=,FRF=,NLG=,ITL=,BEF=,XAU=,XAG=,IEP=
,AUD=,CAD=,ATS=,ESP=,SEK=,NOK=,DKK=,FIM=,PTE=,GRD=

HighlightColourHex = #000000 #FF6300 #FF6300
# The second value on this line sets the background colour of the
# Applet, which is important when the applet is bigger than
# the displayed area.
NormalColourText = Black on white
FontName = ~TimesRoman~ 13 BOLD

AlternatingBGColoursHex =      #FFFFFF #EEEEEE #DDDDDD #EEEEEE
AlternatingBorderColoursHex = #FFFFFF #EEEEEE #DDDDDD #EEEEEE
UpTickColourText = green
DownTickColourText = lightRed
NotickColourText = black

[Header]
Fields = Tick Bid Ask Time

[Tick]
# In quotelists the x-co-ordinate is all that is needed
LabelPosition=60
# A blank label, but the line is needed.
Label =
LabelBundle = Field Label
DataSource = 115
DataBundle = Tick Data

[Bid]
LabelPosition = 80
Label = Bid
LabelBundle = Field Label
DataSource = 22
DataBundle = Field Data

[Ask]
LabelPosition = 150
Label = Ask
LabelBundle = Field Label
```

```
DataSource = 25
DataBundle = Field Data

[Time]
LabelPosition = 220
Label = Time
LabelBundle = Time Label
DataSource = 5
DataBundle = Time Data

[Field Data]
FontName = ~TimesRoman~ 12 BOLD
ColourText = black

[Tick Data]
# We are setting the Cell to display the images but just in case the
# images fail to load we have a backup method using fonts.
Images = true
FontName = ~ZapfDingbats~ 14 BOLD
# Just the font colour needs to be specified because the
# Alternating..ColoursHex parameter will set the background and
border.
ColourText = black

[Field Label]
FontName = ~TimesRoman~ 14 BOLD
ColourText = yellow on black on black

# The rightmost column uses this Label section. This is because we
# want to set the width of this column. Notice that it is the Label
# section which sets this, and not the Data section.

[Time Label]
FontName = ~TimesRoman~ 14 BOLD
ColourText = yellow on black on black
CellWidth = 50

# The time column has smaller text.
[Time Data]
FontName = ~TimesRoman~ 11 NORMAL
ColourText = black
```

An Example Quote (Figure 1.2)

```
[SlingShot]
URL = http://127.0.0.1:80/Reuters/EUR=
HighlightColourHex = #000000 #CCCC00 #CCCC00
NormalColourHex = #FFFFFF #000000

[Header]
Fields = FIELD_115 BID ASK RMF-PAGE FIELD_23 FIELD_26 DEAL FIELD_24
FIELD_27 TIME HIGH LOW

# In quotes (and pages) the LabelPosition and DataPosition
# parameters must have an x co-ordinate followed by a y-co-ordinate.
[FIELD_115]
DataPosition = 0 20
DataSource = 115
DataBundle = Tick

[BID]
LabelPosition = 20 2
Label = Bid
LabelBundle = Field Label
DataPosition = 20 20
DataSource = 22
DataBundle = Bid/Ask Field Data

[ASK]
LabelPosition = 110 2
Label = Ask
LabelBundle = Field Label
DataPosition = 110 20
DataSource = 25
DataBundle = Bid/Ask Field Data

[RMF-PAGE]
LabelPosition = 200 2
Label = RMF Page
LabelBundle = Field Label
DataPosition = 200 20
DataSource = 105
DataBundle = Field Data

[FIELD_23]
```

DataPosition = 20 44
DataSource = 23
DataBundle = Field Data

[FIELD_26]
DataPosition = 110 44
DataSource = 26
DataBundle = Field Data

[DEAL]
LabelPosition = 200 48
Label = Deal
LabelBundle = Field Label
DataPosition = 200 64
DataSource = 78
DataBundle = Field Data

[FIELD_24]
DataPosition = 20 64
DataSource = 24
DataBundle = Field Data

[FIELD_27]
DataPosition = 110 64
DataSource = 27
DataBundle = Field Data

[TIME]
LabelPosition = 200 90
Label = Time
LabelBundle = Field Label
DataPosition = 200 108
DataSource = 5
DataBundle = Field Data

[HIGH]
LabelPosition = 20 90
Label = High
LabelBundle = Field Label
DataPosition = 20 108
DataSource = 203
DataBundle = Field Data

```
[LOW]
LabelPosition = 110 90
Label = Low
LabelBundle = Field Label
DataPosition = 110 108
DataSource = 204
DataBundle = Field Data
```

```
[Tick]
FontName = ~ZapfDingbats~ 16 BOLD
CellWidth = 18
```

```
[Field Label]
FontName = ~Helvetica~ 10 NORMAL
ColourHex = #00FFFF #000000 #000000
```

```
[Bid/Ask Field Data]
FontName = ~SansSerif~ 18 NORMAL
ColourHex = #FFFFFF #000000 #CCCC00
CellWidth = 75
```

```
[Field Data]
FontName = ~SansSerif~ 13 NORMAL
ColourHex = #FFFFFF #000000 #CCCC00
```

An Example Page (Figure 1.4)

Introduction

Requesting pages is slightly different to requesting a single quote. Each line on a page is automatically assigned a DataSource value, increasing from 1 to the number of lines in a page. This is what should be requested. Also, 17 pixels is the amount left between lines in this SLS file. This is the recommended value.

Example

```
[SlingShot]
URL=http://127.0.0.1:80/Reuters/FXFX

HighlightColourText = black on yellow on yellow
ShowRequestBar = true

[Header]
Fields=Line1 Line2 Line3 Line4 Line5 Line6 Line7 Line8 Line9 Line10
Line11 Line12 Line13 Line14

# Only one data section is needed because formatting should be the
# same
# for all lines in the page.

[Default]
FontName = ~Monospaced~ 14 BOLD
CellWidth = 540
ColourText = lightGreen on black on black

[Line1]
DataPosition = 1 1
DataSource = 1
DataBundle = Default

[Line2]
DataPosition = 1 18
DataSource = 2
DataBundle = Default

[Line3]
DataPosition = 1 35
DataSource = 3
DataBundle = Default

[Line4]
DataPosition = 1 52
DataSource = 4
DataBundle = Default
```

```
[Line5]
DataPosition = 1 69
DataSource = 5
DataBundle = Default
[Line6]
DataPosition = 1 86
DataSource = 6
DataBundle = Default
[Line7]
DataPosition = 1 103
DataSource = 7
DataBundle = Default
[Line8]
DataPosition = 1 120
DataSource = 8
DataBundle = Default
[Line9]
DataPosition = 1 137
DataSource = 9
DataBundle = Default
[Line10]
DataPosition = 1 154
DataSource = 10
DataBundle = Default
[Line11]
DataPosition = 1 171
DataSource = 11
DataBundle = Default
[Line12]
DataPosition = 1 188
DataSource = 12
DataBundle = Default
[Line13]
DataPosition = 1 205
DataSource = 13
DataBundle = Default
[Line14]
DataPosition = 1 222
DataSource = 14
DataBundle = Default
```

Changes to SLS Files

What's New

A list of all new SLS file commands is contained below. Details about each new command can be found in the "SLS Parameters – HTML Parameters" section.

- AlternatingBGColours
- AlternatingTextColours
- AlternatingBorderColours
- MergedHighlightColour
- UpHighlightColour
- DownHighlightColour
- UpTickColour
- DownTickColour
- NoTickColour
- ConnectingMessageDataBundle
- CellWidth
- Images
- ShowContributionBar
- ChatApplet
- NoChangeImage
- UpImage
- DownImage
- NoLogging
- NoMerge
- AuthorisationID
- ApiDebug

What's Old

The following Slingshot 1 Client Viewer parameters are no longer supported.

- NextPrev
- BackButton
- AllowRemove
- AllowSorting
- RestoreRequest
- AllowProxy
- RowLabelWidth – This is indirectly supported because quotelist positions are now read and used.
- ShrunkSize
- ShrunkRows
- Overdue
- 3Dbox
- LinkToFID

Creating a Slingshot Ticker

A Ticker applet is also supplied with Slingshot. This ticker allows you to view real-time updates in a scrolling window. The internals of the ticker ensure that only the most recent updates are displayed on screen. Received updates are queued off screen and when it is an object's "turn" to be displayed its most recent data is displayed. Ticker Cells are automatically sized according to the font and string length contained in the cell.

The first point to note is that the Ticker applet uses the "net.Slingshot.Viewer.Ticker.class" file as its start point. A Ticker **cannot** be run from the "net.Slingshot.Viewer.Viewer.class" file.

Example

```
<APPLET CODEBASE="http://Slingshot2Server:80" ARCHIVE="SLSViewer.zip"
CODE="net.Slingshot.Viewer.Ticker.class" WIDTH=400 HEIGHT=34>
<PARAM NAME="UniqueName" VALUE="myFirstTicker">
<PARAM NAME="SRC" VALUE="sls/ticker.sls">
Slingshot Server dead or Java not enabled.
</APPLET>
```

The next important point is that the "DataSource" parameter in Ticker SLS files is different to that of other SLS files. In Tickers the "DataSource" parameter can contain the following comma delimited values.

Value	Description	Example
%n	%n is always	%n, ,3
Number	This specifies the Field ID to be requested. There must be no spaces between this number and any commas.	22, Time ,5
String or Space	Any other characters are displayed "as is" on screen. This makes adding labels and specific numbers of spaces between fields extremely easy.	Bid ,22, Ask ,23

Note: Spaces are parsed when reading the "DataSource" parameter. This means that ", 22," will not request field ID 22. It will place the numerical string 22 on screen with a space before it.

Many features of standard quote/quotelist SLS files are not available to Tickers. Most of these features are quite obvious but will be mentioned here for clarity.

The "Quotelist" and "Record" parameters cannot be displayed by the Ticker Applet. The "Ticker" parameter must be used instead.

All features to do with Highlights are not available. This is because updates are never highlighted in tickers.

The "UpTickColour" and "DownTickColour" parameters are not available in Tickers.

The "RequestBar" parameter is not valid for Tickers.

Label parameters and "CellWidth" parameters are not required because the Ticker Applet handles these features in other ways. The "CellWidth" is calculated automatically, and Labels are entered directly into the "DataSource" parameter so they are not required.



Figure 1.9

An Example Ticker (Figure 1.9)

```
# The Slingshot section contains the general parameters for
# the Slingshot 2 Ticker Applet. All parameters from
# this section can be entered into the <param> section of the HTML
# file. Comments are lines which start with either the
# hash symbol (#) or a semi-colon (;) These lines are not read
# by the applet.
[Slingshot]

# The URL contains details of the Server/server and objects.
# It has the format http://server:port/Service/Object(s).
URL=http://127.0.0.1:80/ReutersSim/ignore

# The ticker details what Objects are to be requested. This
# over-rides any Objects stated in the URL parameter.
Ticker=EUR=, IEP=, JPY=, GBP=, CHF=, NLG=, XAU=
# The server section of the URL
; Server = http://slingshot/

#The port section of the URL. Default is 8000.
; Port = 80

# The Service/Feeder section of the URL
; Feeder=ReutersSFPLUS

# The TickerPauseTime parameter specifies how quickly the ticker
# data moves across the screen. Larger values mean slower movement,
# smaller values mean quicker movement. The default for this is 20
# which is quite smooth but quite slow. A speed of 13 will
# be quicker.
# Smaller values than 13 can be quite CPU intensive.
TickerPauseTime=12
```

```
# This specifies the Default font. It is used if fonts are
# not specified in the dataBundle sections below.
FontName=~TimesRoman~ 10 BOLD

# It is possible to set up a connecting message.
# Note that ConnectingMessage is one word, The default is
# "Connecting to Slingshot Server"
ConnectingMessage = Connecting to Slingshot 2 Server.

# This specifies the default colour for items that are not
# specified in the file. This sets the standard colours.
NormalColourHex = #003167, #DBE3F0, #DBE3F0

# The stale colour is used when a service is marked as dead.
StaleColourText = lightred on black on black

# The Fields line which is used to specify the layout on screen.
# Each item on the fields line represents one line in the ticker.
# The current maximum number of lines is two(2).
Fields = Line1 Line2

# This is a special DataBundle that specifies the format for
# the connecting message. It defaults to red on black, with
# the default font. This cannot be specified in the HTML file.
# Please note the details on Bundle sections below.
# CellWidth is not applicable in this case because it
# automatically sizes itself. No border can be added to it either.
[ConnectingMessageDataBundle]
FontName=~TimesRoman~ 14 BOLDITALIC
ColourHex=#003167, #DBE3F0, #DBE3F0

# Each section contains two parameters
# DataBundle contains the name of the section that defines the
# look of this data. This will be described below.
# In Tickers each DataSource value can contain comma delimited
# sources. %n is a special data source meaning the object name.
# Numbers represent field ID's. These numbers should NOT contain
# any spaces. Any other values, such as labels, will be
# inserted "as is".
```

```
[Line1]
# This data bundle will look similar to this "EUR= 0.9241"
DataSource=%n, ,22
DataBundle=firstLineFormat

# The [Line2] section contains the formatting details for the
# second row in the Quotelist.
[Line2]
# This DataSource parameter will look similar to "Ask 0.9230 15:09"
DataSource=Ask ,23, ,5
DataBundle=secondLineFormat

# This section defines the look of the item whose Bundle it is.
# It contains the following two fields.
# FontName This is the name of the font to be used surrounded
# by tildes. If the font is not available an effort will be made
# to provide a suitable font. This is followed by the point size
# of the font, and this is followed by one of the following
# BOLD, ITALIC, BOLDITALIC, NORMAL. If nothing is here the
# font is NORMAL.
# Colour This is specified as it usually is.
[firstLineFormat]
FontName=~SansSerif~ 16 BOLD
ColourHex=#003167, #DBE3F0, #DBE3F0

[secondLineFormat]
FontName=~SansSerif~ 10 ITALIC
ColourHex=#003167, #DBE3F0, #DBE3F0
```

Appendix A - Specifying Colour Parameters

Introduction

Note: The spelling of the word “colour” in Slingshot is not “color”.

Colours are an important part of any web page, especially so when displaying financial data. For this reason **Slingshot** allows you to specify colours with great precision.

When colours are specified, order is always preserved and is usually as follows

- 1) Text colour
- 2) Background colour
- 3) Border colour

In Figure 1.2 the completely yellow cells are highlighted. The high and low data cells are not highlighted. They are simply surrounded by a border. The up arrow is in the font ZapfDingbats.

The “Alternating...Colour...” parameters take up to 10 colours. These are not in the above form. They are simply a list of colours separated by spaces.

The example below shows a quotelist that uses alternating colours. Its Alternating..Colours... parameters are

```
AlternatingBGColoursHex      = #FFFFFF #EEEEEE #DDDDD #EEEEEE
AlternatingBorderColoursHex = #FFFFFF #EEEEEE #DDDDD #EEEEEE
```

A list of sample colours is available in the “Table of Sample Colours” below.

Creating colours

There are three ways to specify colours, the textual colour markup scheme, Hex colour values and Red-Green-Blue colour values. Each method has a suffix that is added to the standard colour parameter. These are **Text**, **Hex**, and **RGB**. Each method is discussed below.

Example

```
HighlightColourText = black on yellow on yellow
HighlightColourHex  = #000000 #FFFF00 #FFFF00
HighlightColourRGB   = 0,0,0 255,255,0 255,255,0
```

Textual Colour Markup Scheme

In this scheme colours are indicated by their textual names. Only 16 colours are permitted. The suffix is ‘**Text**’.

The general form of the colour specification is as follows:

DataTypeColourText = foreground_Text on background_text on border_text

See the Table of Sample Colours for a list of the 16 available colours.

Hex Colours

This scheme is the same as that used in HTML. Up to 16 million colours may be represented in this scheme. The suffix is '**Hex**'.

The general form of the colour specification is as follows:

DataTypeColourHex = *foreground_Hex background_hex border_Hex*

Hex colour specifications must contain the hash character (#) followed by six hex digits. The hex digits are 0 to 9 and A to F. There should be some spaces between colour definitions and commas are not needed.

RGB Colours

In this scheme, each colour is represented by a set of three values. These values, represented by a three-digit number in the range 000-255, indicate the red, green and blue components of a colour. Up to 16 million colours may be represented in this scheme. The suffix is '**RGB**'.

The general form of the colour specification is as follows:

DataTypeColourRGB = *Red,Green,Blue Red,Green,Blue Red,Green,Blue*

Table of Sample Colours

Textual	Hex Value	RGB Value
Black	#000000	000,000,000
White	#FFFFFF	255,255,255
Yellow	#FFFF00	255,255,000
Red	#800000	128,000,000
LightRed	#FF0000	255,000,000
Green	#008000	000,128,000
LightGreen	#00FF00	000,255,000
Blue	#000080	000,000,128
LightBlue	#0000FF	000,000,255
Cyan	#008080	000,128,128
LightCyan	#00FFFF	000,255,255
Grey	#808080	128,128,128
LightGrey	#C0C0C0	192,192,192
Magenta	#800080	128,000,128
LightMagenta	#FF00FF	255,000,255
Brown	#808000	128,128,000

Figure A3

Appendix B - Specifying Font Parameters

Introduction

The default fonts in Java are dialog Serif, SanSerif, Monospaced and DialogInput but most Java Virtual Machines (in this case your browser) support more fonts. When the **Slingshot** Client Viewer is run the list of available fonts is displayed in the Java Console in your browser. Designers should note that different browsers support different fonts. The Parameter name for fonts is **FontName**.

The general form of the font specification is as follows:

```
FontName = ~NameOfTheFont~ Size Type
```

The equals sign is followed by the Name of the font surrounded by tildes (~). If the specified font name is not available default fonts are used. Labels the font '**Serif**' and data uses the fixed width font '**Monospaced**'. The name is all that is needed to specify a font. A default size of 12 and a type of NORMAL are used in this case. It is better however to specify a size and a format for the font.

The size of the font comes after the font name. It is given as the point size. The default is 12.

The font type follows this. This can be one of the following:

- NORMAL
- ITALIC
- BOLD
- BOLDITALIC

Example

```
FontName = ~ZapfDingbats~ 16 BOLD
```

Up/Down Arrows

There are two ways to add up/down tick arrows. The Images = TRUE method is the first. The images can be seen in **Figure 1.3**.

Example

```
[Tick Data]
Images = TRUE
# Below is just in case the images fail to load.
FontName = ~ZapfDingbats~ 12 BOLD
ColourText = Black on white on white
```

The second method is also demonstrated in the above example. The font, 'ZapfDingbats', can also be used to display an up/down arrow. This only works properly in Internet Explorer and is not supported in Netscape. This can be seen in **Figure 1.2**.

Note: Both of these methods only work with items that have a Data Source of 115 or 14.

Appendix C – Recommended Maximums per Browser

Because of the WDS Servers Merged Update filtering system the Client Viewer is not punished with too many updates per object. This ensures that the dealer sees only the latest updates. See the “Slingshot Administration Guide” for more detail on the Merged Updates feature.

The recommended maximum counts given in the table below are all in relation to one Virtual Machine. Only one connection to the Web Distribution Server (WDS) is maintained per Virtual Machine and this is how performance data is measured.

What is a Virtual Machine?

In the simplest of terms a Virtual Machine (VM) is where Java code is run. A Virtual Machine is a software program, mimicking a hardware machine. Most Internet browsers contain Virtual Machine's for running Java Applets. There are also other types of Virtual Machine, which do not run in Internet browsers. This is explained in the section “Running the Slingshot Client Viewer as an application”.

Depending on your browser, and how you start it you will get a different number of Virtual Machines. In Netscape you always have a maximum of one Virtual Machine running. In Internet Explorer when you open a new Browser (via the Start Menu for example) you get a new Virtual Machine. But if you open a new window via File | New | Window the same Virtual Machine is reused. The true maximums for these figures depend on the hardware that the Client Viewer is running on. Obviously, in an Internet environment the specification of user machines is impossible to control. Within an Intranet however this is easier so the Designer can take this into consideration.

Recommended Maximum Applets Per VM	=	10
Recommended Maximum Objects/Records Per VM	=	250

Appendix D - Browser Versions

The following table outlines the recommended browsers for running the Slingshot Client Viewer. Internet Explorer 5.5 with Service Pack 1 and Netscape 4.76 are the most highly recommended browsers. There are no known problems with Netscape 6 but this has not been sufficiently tested.

Browser	Operating System	Opinion	Comments
IE 5.5 SP1	NT 4.0	Recommended	
IE 5.5	NT 4.0	Not Recommended	Hitting back in a page with a Java Applet sometimes results in the browser crashing
IE 5.0	NT 4.0	No known problems	
IE 4.0	NT 4.0	No known problems	Not fully tested
Netscape 6.0	NT 4.0	No known problems	Not fully tested
Netscape 4.76	NT 4.0	Recommended	Resizing the Window results in the Client Viewer being reloaded. This is a known Netscape bug.
Netscape 4.74	NT 4.0	Not Recommended	Upgrade to 4.76
Netscape 4.73	Solaris	Not Recommended	Upgrade to 4.76
Netscape 4.7i	AIX 4.3	Not Recommended	When you resize the Netscape window the applets no longer update. You need to do a "Hard Refresh" to fix this.

Abbreviations :

IE = Internet Explorer

SP = Service Pack

Appendix E - Client Hardware Machine Specifications

The following table outlines the minimum and recommended specifications for machines running the Slingshot Client Viewer.

Client Hardware	CPU	Memory
Minimum	Pentium 133	32Mb
Recommended	P II 400 or greater	128Mb
Minimum	Sparc 20	64Mb
Recommended	Sun Ultra 5	256Mb

Appendix F - Frequently Asked Questions

How do I debug the Client Viewer?

Debugging of the Client Viewer is different in Internet Explorer and Netscape.

In Netscape 4.X select "Communicator | Tools | Java Console".

In Internet Explorer 4.X or 5.X you must first ensure that the Java Console is switched on. To do this go to "Tools | Internet Options". Select the "Advanced" tag and scroll down to "Microsoft VM". Ensure that "Java console enabled (requires restart)" is ticked and if required restart the browser. Now to view debugging information simply go to "View | Java Console".

The Java Console in both browsers shows more details about how the Slingshot Client Viewer is performing. Exceptions, which are errors, will appear in the console detailing where the problem is. This should help you to debug both SLS files and when you are setting up the Slingshot WDS.

What if the Client Viewer does not load properly?

If the Client Viewer does not load at all it means that the Slingshot WDS is not functioning. In this case please contact the web site administrator.

If however, there is a problem with the Client Viewer and it is displaying either its connecting message, "Connecting to Slingshot Server..." or you get a "Requester is Closed" message for more than thirty seconds a "Hard Refresh" is needed.

What is a "Hard Refresh"?

A hard refresh is when you force the browser to reload everything on a web page. There is no way to guarantee a full refresh if you are going through a proxy.

How do I do a "Hard Refresh" in Internet Explorer?

Hold the 'Ctrl' key at the same time as pressing the Refresh button with your mouse.

How do I do a "Hard Refresh" in Netscape?

Hold the 'Shift' key at the same time as pressing the Reload button with your mouse.