webRDP Guide

© 2011 Stoneware, Inc.

Note:
This document is the property of webInflection. It is not to be copied, reproduced, or printed without prior consent from Stoneware, Inc.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Part I  Introduction</strong></td>
<td>3</td>
</tr>
<tr>
<td>1 Welcome</td>
<td>3</td>
</tr>
<tr>
<td>2 Overview</td>
<td>4</td>
</tr>
<tr>
<td>3 New in 1.0.5</td>
<td>5</td>
</tr>
<tr>
<td>4 Requirements</td>
<td>6</td>
</tr>
<tr>
<td><strong>Part II  Features</strong></td>
<td>7</td>
</tr>
<tr>
<td>1 Overview</td>
<td>7</td>
</tr>
<tr>
<td><strong>Part III  Stand Alone Client</strong></td>
<td>12</td>
</tr>
<tr>
<td>1 Overview</td>
<td>12</td>
</tr>
<tr>
<td>2 Deployment</td>
<td>20</td>
</tr>
<tr>
<td><strong>Part IV  Embedded Client</strong></td>
<td>23</td>
</tr>
<tr>
<td>1 Overview</td>
<td>23</td>
</tr>
<tr>
<td>2 HTML Deployment</td>
<td>25</td>
</tr>
<tr>
<td>3 JSP Deployment</td>
<td>27</td>
</tr>
<tr>
<td>Index</td>
<td>0</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Welcome

The webRDP client is a Java-based client designed to communicate with backend Microsoft Terminal Servers and remote Windows desktops using RDP (remote desktop protocol). The client was specifically designed to improve the performance, reliability, and functionality of remote desktop connectivity on Windows and non-Windows platforms such as Linux and Mac OS X. This document will describe how to start, configure, and embed the webRDP client into your web environment.
1.2 Overview

The webRDP client is a new RDP client designed from the "ground up". The client was written in Sun's Java programming language to provide maximum portability across a variety of operating system platforms. The webRDP client runs as a stand alone Java application or an embedded Java applet within a web page or web application.

As a stand alone application, the user can execute the client on any operating system platform that supports the Sun Java Virtual Machine. These operating systems would include Linux (e.g. - Redhat, SuSE, Ubuntu, etc.), Windows (e.g. - XP, Vista, 7, etc.), and Mac OS X (e.g. - 10, 10.5, etc.).

As an embedded applet within another web page or web application, the webRDP client can be invoked as long as the browser has the Sun Java plug-in enabled. The user can view the remote desktop or terminal session within the browser window or within an Iframe inside a web page. This document will describe how the stand alone client is invoked and how the webRDP client can be embedded into other applications.
1.3 New in 1.0.5

The webRDP 1.0.5 client supports the following new features:

- Basic webRDP Client Features
  - Cut and Paste
  - RDP Compression
  - Resolution Slider and enhanced resolutions

- Advanced webRDP Client Features
  - Cut and Paste
  - RDP Compression
  - Resolution Slider and enhanced resolutions
  - HTTP(S) Proxy Support
  - SOCKS 5 Support
  - Customizable GUI Banner
  - Customizable Splash Screen
  - Connection Broker Support (session select)
1.4 Requirements

Below are the client-side requirements for both the stand alone and embedded versions of the webRDP client:

**Stand Alone**

- Sun JRE (Java Runtime Environment) versions 1.5.x, and 1.6.x

**Embedded**

- Sun Java plug-in versions 1.5.x and 1.6.x
- Firefox, Internet Explorer, Chrome, Safari, Opera
2 Features

2.1 Overview

A matrix of webRDP features is displayed below.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Basic Client</th>
<th>Advanced Client</th>
<th>OEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username/Password</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Display Settings</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Start Programs</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RDP 5 Bulk Compression</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Experience</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Keep Alive</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cut and Paste</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Customizable Banner</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Customizable Splash</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>HTTP(S) Proxy Support</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SOCKS 5 Support</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Connection Broker Support</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Encrypted Password</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

A description of each webRDP feature is listed below.

- **Computer (required)**

  The IP address, DNS name, or machine name of the backend device the webRDP client will attempt to connect to. The user can pass a non-standard port number (i.e. - something other than the default RDP port 3389) by appending ":[port number]" to the computer name.

  *Example: mymachine.stone-ware.com:4595

- **Username**

  The user ID required to access the device via the remote desktop protocol (RDP). This parameter is not required. The remote desktop or Terminal Server will prompt if authentication is required.

- **Display**

  The Display tab controls the screen resolution displayed through the webRDP client. The client supports multiple display settings including:
  - 640x480
  - 800x600
• 1024x768
• 1280x1024
• 1440x900
• 1600x1200
• 1680x1050
• 1920x1080

- Start Program

The Start Program feature allows the webRDP client to automatically start a specific application on the backend Terminal Server. The Terminal Server will suppress the desktop view and only display the application's interface via the connection. This feature is only available when connecting to a Microsoft Terminal Server.

- Start Folder (optional)

  Specifies the working directory for the application specified in the Start Program field.

- RDP Compression

  RDP compression is enabled within the webRDP client to optimize performance between the client and the backend Terminal Server or remote desktop. The compression is enabled by default and cannot be disabled on the client. If the host does not support RDP 5 bulk compression, the client will default back to RDP 4 compression.

- Experience

  The experience tab will control some of the performance settings associated with controlling the remote desktop or terminal session. Performance between the webRDP client and remote desktop can be improved by disabling graphical features of the remote desktop when creating the RDP session. The performance settings are described below:

  • Desktop Background - When checked, the remote desktop or terminal session will display any desktop background configured on the machine. Often the desktop background can be very graphic intensive and require the client to redraw the screen more frequently, thus slowing performance.

  • Font Smoothing - When checked, the webRDP client will display font smoothing if enabled on a 2003 Terminal Server or a 2008 Terminal Server. This feature will improve the visual presentation of desktop windows and text.

  • Desktop Composition (Vista and Windows 7 feature) - When Desktop Composition is enabled, individual windows no longer draw directly to the screen or primary display device as they did in earlier versions of Windows. Instead, their drawing is redirected to off-screen surfaces in video memory, which are then rendered into a desktop image and presented on the display. This feature will consume more bandwidth and can be
disabled by removing the check from the box.

- Show Contents of Window While Dragging - When checked, the webRDP client will display the contents of the window being dragged across the desktop interface. When disabled, the contents of the window are suppressed while being moved within the desktop.

- Menu Animation - When checked, the menu animations of the desktop will be displayed on the remote desktop client. These animations will generate more network traffic when being sent to the remote desktop client (i.e. -webRDP). Menu animations can be disabled to reduce communication traffic by disabling the check box.

- Themes - When checked, the theme associated with the desktop will be displayed in the webRDP client. Themes are more graphically intensive and therefore will generate more traffic between the webRDP client and remote desktop and/or Terminal Server. To disable the theme within the remote desktop client, uncheck the box.

  - Keep Alive

    The keep alive option will send a packet to the backend Terminal Server or desktop on a predetermined interval to stop the device from being disconnected by networking gear (switches, routers, etc.) due to inactivity.

  - Cut and Paste (Clipboard)

    The webRDP client supports the transfer of text between the remote desktop and the client using the clipboard feature of the operating system.

  - Customize Banner (Stand-alone Mode)

    Allows an administrator to change the default graphic banner displayed at the top of the webRDP user interface. This feature allows OEMs to replace the shipping graphic banner with their own branded graphic banner.

  - Customize Splash Screen (Embedded Client)

    Allows the administrator to change the default graphic splash screen that is displayed when the webRDP client is invoked within a web page or web application.

  - Proxy Support

    The webRDP advanced client supports the use of HTTP(S) Forward Proxy servers. When enabled, the webRDP client will make requests to the backend Terminal Server or remote desktop through a designated Proxy. This feature is useful for organizations who do not wish to expose their internal Terminal Servers or remote desktops to the Internet.
The proxy server's IP Address and Credentials (username/password) can be passed as part of the RDP connection. Once the connection to the proxy server is accepted, all RDP traffic will be redirected to the internal Terminal Server or remote desktop.

- Chaining - feature allowing two or more proxy servers to be defined when connecting to a Terminal Server or remote desktop.

- SOCKS Support

The webRDP advanced client supports the use of SOCKS 5 servers. When enabled, the webRDP client will make requests to the backend Terminal Server or remote desktop through a designated SOCKS 5 server. This feature is useful for organizations who do not wish to expose their internal Terminal Servers or remote desktops to the Internet. The SOCKS server's IP Address and Credentials (username/password) can be passed as part of the RDP connection. Once the connection to the SOCKS server is accepted, all RDP traffic will be redirected to the internal Terminal Server or remote desktop.

- Chaining - feature allowing two or more SOCKS servers to be defined when connecting to a Terminal Server or remote desktop.

*Note - use of clear text username/password is currently supported with SOCKS 5*

**Did You Know:**

You can chain a combination of SOCKS and PROXY servers together. The webRDP client can connect through a HTTPS proxy server and then through a SOCKS server to get to a backend RDP session.

- Session Select (Connection Broker)

Session select is a feature by Microsoft designed to connect to an RDP session through a connection broker. The connection broker will take information from the connection
string and make the connection to a backend RDP session. Session selects version 1 and 2 as described below:

1. Version 1 - ID (32bit integer)
2. Version 2 - ID (32bit integer) and String

- Encrypted Password

New feature that allows the webRDP client to pass an encrypted password between the client and the backend RDP session. The encryption method supports RSA 512 Public Key/Private Key format. The feature works in stand-alone mode or embedded within a web page/application. In addition to encrypting the user's password, the administrator can set an expiration on the password so that it is no longer valid after a given number of hours.
3 Stand Alone Client

3.1 Overview

When executed in stand alone mode, the webRDP client will display the user configuration interface (seen below). The interface allows the end user to configure the backend device, display resolution, startup programs, and performance settings. Each of these feature categories is represented by a tab at the top of the interface. The tabs and their purpose are described in the section below.

General

The General tab allows the user to specify the backend Terminal Server or desktop the webRDP client will be connecting to. The user will be required to enter the computer name, IP address, or DNS name in order to connect to any backend system.

- Computer (required)
The IP address, DNS name, or machine name of the backend device the webRDP client will attempt to connect to. The user can pass a non-standard port number (i.e. - something other than the default RDP port 3389) by appending ":[port number]" to the computer name.

Example: mymachine.stone-ware.com:4595

- Username

The user ID required to access the device via the remote desktop protocol (RDP). This parameter is not required. The remote desktop or Terminal Server will prompt if authentication is required.

Display

The Display tab controls the screen resolution displayed through the webRDP client. The user can select the "slider" to pick between the various screen resolutions.

The client supports multiple display settings including 640x480, 800x600, 1024x768, and 1280x720, 1280x1024, 1440x900, 1600x1200, 1680x1050, and 1920x1080.

Programs

This feature is only available when connecting to a Microsoft Terminal Server. The programs tab allows the webRDP client to automatically start a specific application on the backend.
Terminal Server. The Terminal Server will suppress the desktop view and only display the application's interface via the connection.

- Start the Program on Connection

  When checked, the webRDP client will direct the Terminal Server to start the application specified on the Program Path / File Name setting. When disabled, the request from the webRDP client is to view the entire desktop.

- Program Path and File Name

  The file system path to the application that will be executed remotely on the Microsoft Terminal Server. The path should be relative to the Terminal Server (i.e. - the Terminal Server's local file system) and include the executable's name.

  Example: c:\Program Files\Microsoft\Office 11\excel.exe

- Start Folder (optional)

  Specifies the working directory for the application

Experience

The experience tab will control some of the performance settings associated with controlling the remote desktop or terminal session. Performance between the webRDP client and remote desktop can be improved by disabling graphical features of the remote desktop when creating the RDP session. The performance settings are described below:

- Desktop Background

  When checked, the remote desktop or terminal session will display any desktop background configured on the machine. Often the desktop background can be very graphic intensive and require the client to redraw the screen more frequently, thus slowing performance.

- Font Smoothing

  When checked, the webRDP client will display font smoothing if enabled on a 2003 Terminal Server or a 2008 Terminal Server. This feature will improve the visual presentation of desktop windows and text.

- Desktop Composition (Vista and Windows 7 feature)

  When Desktop Composition is enabled, individual windows no longer draw directly to the screen or primary display device as they did in earlier versions of Windows.
Instead, their drawing is redirected to off-screen surfaces in video memory, which are then rendered into a desktop image and presented on the display. This feature will consume more bandwidth and can be disabled by removing the check from the box.

- **Show Contents of Window While Dragging**

  When checked, the webRDP client will display the contents of the window being dragged across the desktop interface. When disabled, the contents of the window are suppressed while being moved within the desktop.

- **Menu Animation**

  When checked, the menu animations of the desktop will be displayed on the remote desktop client. These animations will generate more network traffic when being sent to the remote desktop client (i.e. -webRDP). Menu animations can be disabled to reduce communication traffic by disabling the check box.

- **Themes**

  When checked, the theme associated with the desktop will be displayed in the webRDP client. Themes are more graphically intensive and therefore will generate more traffic between the webRDP client and remote desktop and/or Terminal Server. To disable the theme within the remote desktop client, uncheck the box.

- **Keep Alive**

  The keep alive option will send a packet to the backend Terminal Server or desktop on a predetermined interval to stop the device from being disconnected. The user can set the keep alive by checking the box and selecting the number of seconds the webRDP client should send a keep alive packet.

**Banner (Advanced webRDP Client Feature)**

The default banner that is displayed at the top of the webRDP user interface can be customized in the advanced client. This feature is for OEMs that wish to customize the interface as part of another product offering. To customize the interface, follow the steps below:

- **Before You Begin**
  - Your banner must be in JPG format
  - Your banner should be approximately 400x70 pixels in size
  - Your banner must be named `oem_banner.jpg`

To replace the default webRDP user interface banner follow the steps below:
1. Copy your new banner file (oem_banner.jpg) into the same file system directory as the webRDP.jar file.

The webRDP.jar file should pickup the new banner and display it inside the user interface.

**Configuring a Proxy Server (Advanced webRDP Client Feature)**

To connect to a HTTP or HTTPS forward proxy server the user should know the following information:

- **Before You Begin**
  - IP Address or DNS name of the forward proxy server
  - Username and password for the forward proxy (if necessary)
  - You must still configure the destination remote desktop or Terminal Server on the General tab

- **Connecting to a Proxy server**

To configure the webRDP client to connect to a remote desktop or Terminal Server via an HTTP forward proxy, follow the steps below:

1. From the webRDP interface, select the NETWORK tab
2. Check the box next to PROXY CONNECTION
3. Select the ADD button at the bottom of the screen
4. Enter the proxy server’s information
Proxy type - select the type of proxy (HTTP, HTTPS, or SOCKS 5)
Proxy Address - IP Address or DNS name of the proxy server
Port - the port number of the HTTP(S) proxy (default is 8080)
User - a username to authenticate against the proxy server (if required)
Password - a password for the username sent to the proxy server

5. Select the OK button
6. Continue configuring the connection to the backend Terminal Server or remote
desktop through the other tabs (e.g. - general, display, and programs)

When configured correctly, the webRDP client will connect to the proxy server using the
information provided and request the proxy server to communicate with the backend
remote desktop or Terminal Server. The proxy server will send all RDP requests to the
Computer specified on the General tab.

- Chained Proxies

Proxy chaining is the process of utilizing two or more proxies to connect to the
backend Terminal Server or remote desktop. The user can enter additional proxies (as
described above) and then specify the order that the client should connect through
the proxy servers using the UP and DOWN buttons. The user can mix the types of
proxy servers in the chain selecting between HTTP, HTTPS, and SOCKS 5.

- SOCKS Server

Configuring the client to communicate with a SOCKS 5 server is very similar to configuring
the proxy connection. To configure the webRDP client to connect to a remote desktop or
Terminal Server via an SOCKS 5 server, follow the steps below:
1. From the webRDP interface, select the NETWORK tab
2. Check the box next to PROXY CONNECTION
3. Select the ADD button at the bottom of the screen
4. Enter the SOCKS server’s information

Proxy type - select the type of proxy (HTTP, HTTPS, or **SOCKS 5**)
Proxy Address - IP Address or DNS name of the proxy server
Port - the port number of the SOCKS server
User - a username to authenticate against the SOCKS server (if required)
Password - a password for the username sent to the SOCKS server

7. Select the OK button
8. Continue configuring the connection to the backend Terminal Server or remote
desktop through the other tabs (e.g. - general, display, and programs)

When configured correctly, the webRDP client will connect to the SOCKS server using the
information provided and request the SOCKS server to communicate with the backend
remote desktop or Terminal Server. The SOCKS server will send all RDP requests to the
Computer specified on the General tab.

- Chained Proxies

Proxy chaining is the process of utilizing two or more proxies to connect to the
backend Terminal Server or remote desktop. The user can enter additional SOCKS
servers (as described above) and then specify the order that the client should connect
through the SOCKS servers using the UP and DOWN buttons. The user can mix the
types of proxy servers in the chain selecting between HTTP, HTTPS, and SOCKS 5.
3.2 Deployment

The webRDP client can be started in stand-alone (GUI Interface) mode by any of the following methods:

- Double-clicking the webRDP application

  With the Java Runtime Environment (JRE) installed on the desktop, the user should be able to double-click on the webRDP.jar file and have it automatically start on the desktop. If the application does not automatically start, check the installation of the JRE on the local machine.

- Starting the webRDP application from a command prompt

  The user can start the webRDP client from the command line by typing the following command:

  \[ java -jar webRDP.jar \]

  The user can start the webRDP client using a set of command-line parameters. To view the available command-line parameters start the webRDP client with the /? option. See below:

  \[ java -jar webRDP.jar /? \]
the screen above is displayed showing all of the currently supported command-line parameters

Starting the client from a command prompt or terminal session allows the user to pass optional parameters that would be set from within the interface. An example of starting the webRDP client with parameters is displayed below:

java -jar webRDP.jar /v:mymachine.stone-ware.com

java -jar webRDP.jar /v:mymachine.stone-ware.com /u:admin /p:letme1n

A list of optional parameters are provided below:

- /v:[value] = server name and/or port number
- /u:[value] = username
- /p:[value] = user's password
- /t:[value] = startup program
- /wh:[value] = aspect ratio (width and height)
- /w:[value] = screen width
- /h:[value] = screen height
- /ka:[value] = number of seconds between keep alive events (10 - 300)
- /pf:[value] = value for the performance flags (add the values below)

Disable Wallpaper: 1
Disable Full Window Drag: 2
Disable Menu Animations: 4
Disable Theming: 8
Disable Cursor Shadow: 32
Disable Cursor Settings: 64

Example: a setting of 96 would mean Cursor Shadow and Cursor Settings were disabled.

☐ A list of advance webRDP parameters are provided below:

- /ep:[password value] = encrypted user password

Use the PassGen utility that ships with the advanced client to create the encrypted password. To create an encrypted password, follow the steps below:

1. copy the webRDP.jar, license, and PassGen.jar file into the same directory
2. Run the PassGen utility at a command line (i.e. - java - jar PassGen.jar)
3. Enter the password to be encrypted and hit ENTER
4. Enter the FROM date (0 is a perpetual password that does not expire) and hit ENTER
5. An encrypted password will be displayed. Copy the password onto the /ep parameter.

- /sv:[version number] = version number of session selection method
  (supports either 1 or 2)
- /sid:[connection ID] = the connection ID (must be an integer)
- /sstr:[connection string] = string that will be passed to the connection broker (version 2 only)
- /pr:[<proxy>,<proxy>] = proxy server used to connect to Terminal Server or remote desktop

Proxy format = (http | https | socks )://user:
password@address:[port]
Example = https://fred:
abcpassword@187.2.33.4:8080
4 Embedded Client

4.1 Overview

The webRDP client was specifically designed to be embedded within other web applications and web pages. Any developer can invoke the webRDP client from their web application or web page by calling the applet and passing the appropriate parameters. The applet can be invoked within the web page by using the standard applet command:

```xml
<applet name='rdp' code='com.webinflection.webrdp.MainApplet' archive='webRDP.jar' width='100%' height='100'>

Parameters can be passed by adding the attributes between the applet tags:

```xml
<param name='username' value='<%= user %>'>

The parameters that can be passed to the applet tag are listed below:

- `<host>` = host IP address or DNS name
- `<port>` = host RDP port (default is 3389)
- `<username>` = username
- `<password>` = user’s password
- `<domain>` = Terminal Server Domain (optional)
- `<program>` = program automatically launched (Terminal Server only)
- `<directory>` = set programs working directory (optional)
- `<maxwidth>` = desktop screen maximum width in pixels
- `<maxheight>` = desktop screen maximum height in pixels
- `<bpp>` = desktop color depth (8/15/16/24)
- `<keepalive>` = number of seconds between keep alive events (10-300)
- `<pf>` = performance flags (add the values to apply multiple performance flags)

Disable Wallpaper: 1
Disable Full Window Drag: 2
Disable Menu Animations: 4
Disable Theming: 8
Disable Cursor Shadow: 32
Disable Cursor Settings: 64

Example: a setting of 96 would mean Cursor Shadow and Cursor Settings were disabled.

The advanced parameters that can be passed to the applet tag are listed below:

- `<epassword>` = encrypted user password

- `<ss-version>` = version number of session selection method (supports either 1
or 2)

- `<ss-id>=` = the connection ID (must be an integer)
- `<ss-string>=` = string that will be passed to the connection broker (version 2 only)
- `<proxy>=` = proxy server used to connect to Terminal Server or remote desktop

Proxy format = (http | https | socks)://user:password@address: [port]

Example = https://fred:abcpassword@187.2.33.4:8080

To see full samples, review the HTML Deployment or JSP Deployment sections in this document.
4.2 HTML Deployment

This section of the guide is to provide an example for embedding the webRDP client into an HTML page that runs inside the web browser. Before you begin, you will need to follow the steps below to deploy the webRDP client jar file.

1. Copy the webRDP.jar file to the HTML docs directory of the web server.
2. Create an HTML page that invokes the webRDP applet.
3. Add the parameters that should be passed into the applet when it starts (e.g. - username, password, applications, screen resolution, etc.)

A sample of an HTML is given below:

```html
<html>
<head>
  <title>webRDP&amp;#0153;</title>
  <meta http-equiv='content-type' content='text/html; charset=iso-8859-1'>
  <meta http-equiv='content-style-type' content='text/css'>
  <meta http-equiv='expires' content='Wed, 26 Feb 1997 08:21:57 GMT'>
  <meta http-equiv='pragma' content='no-cache'>
  <!-- Sets margin around applet -->
  <style>
    body.swproxyBody { margin:4px; }
  </style>
  <script type='text/javascript'>
    // This method is called after the user logs out of their RDP session. The method name is a configurable applet parameter.
    function rdpOnLogout() {
      alert ('User has Logged out');
    }
  </script>
</head>

<body class='swproxyBody'>
  <!-- Loads the applet and utilizes 100% of browser window width and height. Width and height could be hard coded to specific values -->
  <applet name='rdp' code='com.webinflection.webrdp.MainApplet' archive='webRDP.jar' width='100%' height='100%'>
    <!-- Hostname or IP Address of Terminal Server -->
    <!-- This is a required parameter -->
    <param name='host' value='10.1.1.25'>
</body>
</html>
```
<param name='port' value='3389'>

<param name='username' value='joe'>

<param name='password' value='cat1dog2'>

<param name='domain' value=''>

<param name='program' value='c:\windows\system32\notepad.exe'>

<param name='directory' value='c:\windows\system32'>

<param name='onlogout' value='javascript:rdpOnLogout();'>

</html>
4.3 JSP Deployment

This section of the guide is to provide an example for embedding the webRDP client into an JSP (Java Server Page) that runs inside the web browser. Before you begin, you will need to follow the steps below to deploy the webRDP client jar file.

1. Copy the webRDP.jar file to the HTML docs directory of the web server.
2. Create a Java Server Page that invokes the webRDP applet
3. Add the parameters that should be passed into the applet when it starts (e.g. - username, password, applications, screen resolution, etc.)

A sample of JSP page is given below:

```jsp
<%@page import="java.net.*"%>
<%

// This example modifies the HTML example and uses the JSP to pass the parameters to the applet
// We are not really dynamically creating the parameters, but this is where you would make calls to your configuration apis to get the parameter values

String host = "10.1.1.25";
String port = "3389";

// Single Sign-on Parameters.

String user = "administrator";
String pass = "stoneware";
String domain = "";

// Published Application Parameters. Remember to URLEncode the program and the directory

String program = URLDecoder.decode("c:\\windows\\system32\\notepad.exe" );
String directory = URLDecoder.decode("c:\\windows\\system32\"");

%>

<!--

This example demonstrates RDP SSO ( Single Sign-on ) and a Published Application ( MS notepad in this case )

-->```

```jsp
<html>
<head>
```
<title>webRDP</title>
<meta http-equiv='content-type' content='text/html; charset=iso-8859-1'>
<meta http-equiv='content-style-type' content='text/css'>
<meta http-equiv='expires' content='Wed, 26 Feb 1997 08:21:57 GMT'>
<meta http-equiv='pragma' content='no-cache'>

<!-- Sets margin around applet -->
<style>
body.swproxyBody { margin:4px; }
</style>

<script type='text/javascript'>
// This method is called after the user logs out of their RDP session. The method name is a configurable applet parameter.

function rdpOnLogout() {
    alert ('User has Logged out ');
}
</script>
</head>
<body class='swproxyBody'>

<!-- Loads the applet and utilizes 100% of browser window width and height. Width and height could be hard coded to specific values -->
<applet name='rdp' code='com.webinflection.webrdp.MainApplet' archive='webRDP.jar'
width='100%' height='100%'>

<!-- Hostname or IP Address of Terminal Server -->
<!-- This is a required parameter -->
<param name='host' value='<%= host %>'

<!-- Port that the Terminal Server -->
<!-- This is a required parameter -->
<param name='port' value='<%= port %>'

<!-- Username to authenticate to Terminal Server with -->
<!-- Optional SSO Parameter -->
<param name='username' value='<%= user %>'

<!-- Password to authenticate to Terminal Server with -->
<!-- Optional SSO Parameter -->
<param name='password' value='<%= pass %>'

<!-- AD Domain name to authenticate to Terminal Server with -->
<!-- Optional SSO Parameter -->
<param name='domain' value='<%= domain %>'
</applet>
</body>
<!-- Application to start. This value should be url encoded. -->
<param name='program' value='%= program %'>

<!-- Working directory for Application. This value should be url encoded. -->
<param name='directory' value='%= directory %'>

<!-- This specifies a javascript method to be called after the user logs out of the RDP session -->
<param name='onlogout' value='javascript:rdpOnLogout();'>

</applet>
</body>
</html>