

Document name	Targeting Different device densities in android devices
Version no.	1.0
Release date	15-Mar-2012

This document of Cybage Software Pvt. Ltd. is for restricted circulation. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means – recording, photocopying, electronic and mechanical, without prior written permission of Cybage Software Pvt. Ltd.

Targeting Different device densities in android devices

Table of Contents

1. Introduction	3
2. Targeting Different device densities in android devices	3
2.1 Introduction.....	3
2.2 Targeting Device Densities With CSS	4

1. Introduction

This will be a reference document for the Targeting Different device densities in android devices.

2. Targeting Different device densities in android devices

2.1 Introduction

The density of a device's screen is based on the screen resolution, as defined by the number of dots per inch (dpi). There are three screen density categories supported by Android: low (ldpi), medium (mdpi), and high (hdpi). A screen with low density has fewer available pixels per inch, whereas a screen with high density has more pixels per inch (compared to a medium density screen). The Android Browser and WebView target a medium density screen by default.



Figure 2 A web page with viewport width=device-width and target-densitydpi=device-dpi.

Because the default target density is medium, when users have a device with a low or high density screen, the Android Browser and WebView scale web pages (effectively zoom the pages) so they display at a size that matches the perceived appearance on a medium density screen. More specifically, the Android Browser and WebView apply approximately 1.5x scaling to web pages on a high density screen (because its screen pixels are smaller) and approximately 0.75x scaling to pages on a low density screen (because its screen pixels are bigger).

For example, to prevent the Android Browser and WebView from scaling your web page for different screen densities, set the target-densitydpi viewport property to device-dpi. When you do, the page is not scaled. Instead, the page is displayed at a size that matches the current screen's density. In this case,

you should also define the viewport width to match the device width, so your web page naturally fits the screen size.

```
<meta name="viewport" content="target-densitydpi=device-dpi, width=device-width" />
```

2.2 Targeting Device Densities With CSS

Method-I

The Android Browser and WebView support a CSS media feature that allows you to create styles for specific screen densities—the `-webkit-device-pixel-ratio` CSS media feature. The value you apply to this feature should be either "0.75", "1", or "1.5", to indicate that the styles are for devices with low density, medium density, or high density screens, respectively.

For example, you can create separate stylesheets for each density:

```
<link rel="stylesheet" media="screen and (-webkit-device-pixel-ratio: 1.5)"
href="hdpi.css" />
<link rel="stylesheet" media="screen and (-webkit-device-pixel-ratio: 1.0)"
href="mdpi.css" />
<link rel="stylesheet" media="screen and (-webkit-device-pixel-ratio: 0.75)"
href="ldpi.css" />
```

Method-II

Or, specify the different styles in one stylesheet:

```
#header {
    background:url(medium-density-image.png);
}
@media screen and (-webkit-device-pixel-ratio: 1.5) {
    /* CSS for high-density screens */
    #header {
        background:url(high-density-image.png);
    }
}
@media screen and (-webkit-device-pixel-ratio: 0.75) {
    /* CSS for low-density screens */
    #header {
        background:url(low-density-image.png);
    }
}
```

Method-III

Target the device density with JavaScript

The Android Browser and WebView support a DOM property that allows you to query the density of the current device—the `window.devicePixelRatio` DOM property. The value of this property specifies the scaling factor used for the current device. For example, if the value of `window.devicePixelRatio` is "1.0",

then the device is considered a medium density device and no scaling is applied by default; if the value is "1.5", then the device is considered a high density device and the page is scaled 1.5x by default; if the value is "0.75", then the device is considered a low density device and the page is scaled 0.75x by default. Of course, the scaling that the Android Browser and WebView apply is based on the web page's target density—as described in the section about Defining the viewport target density, the default target is medium-density, but you can change the target to affect how your web page is scaled for different screen densities.

For example, here's how you can query the device density with JavaScript:

```
if (window.devicePixelRatio == 1.5)
{
    alert("This is a high-density screen");
}
else if (window.devicePixelRatio == 0.75)
{
    alert("This is a low-density screen");
}
```

Reference:

<http://developer.android.com/guide/webapps/targeting.html>