Standard Advertsising Formats

	Format size in px (width x height)	size in kb	File formats	available on
Billboard [1]	980 x 250	100	jpg, gif, HTML5	Comdirect.de/Informer*, onvista.de
Medium Rectangle [1]	300 x 250	80	jpg, gif, HTML5	Comdirect.de/Informer*, onvista.de
Halfpage Ad [1]	300 x 600	80	jpg, gif, HTML5	onvista.de
Skyscraper [1]	max. 200 x 600	80	jpg, gif, HTML5	onvista.de
Sitebar [2]	mind. 120 x 600 (stick to proportion)	150	Javascript Redirect	onvista.de

Exclusive Placements

	Format size in px (width x height)	size in kb	File formats	available on
Homepage Exclusive Billboard [1], [3]	980 x 250	100	jpg, gif, HMTL5	onvista.de
Homepage Exclusive First Contact Billboard [1], [3]	980 x 250	100	jpg, gif, HMTL5	onvista.de

Contact sales onvista media
sales@onvista.de

E-Mail Marketing

	Format size in px (width x height)	size in kb	File formats	available on
Standalone Newsletter [4]	see description [4]	60	plain text, html	onvista.de, Cash/onvista

Content Integration

	Format size in px (width x height)	size in kb	File formats	available on
Premium-Box onvista	**Headline:** max. 35 characters incl. spaces **Teaser Text:** 3-4 Lines with max. 45 characters incl. spaces each **Teaser Box Image:** 100 x 100	_	jpg, gif	onvista.de
Premium-Box my onvista	**Headline:** max. 30-40 characters incl. spaces **Teaser-Text:** ca. 260 characters incl. spaces (+ link) **Teaser-Box Image:** 3:1 Format **Logo:** possible	_	eps	onvista.de
Premium-Box Informer	**Headline**: max. 30 characters incl. spaces **Teaser-Text:** max. 145 characters incl. spaces (+ link) **Teaser-Box Image**: 310 x 130	_	jpg, gif	Comdirect.de/Informer
Native Ad	**Teaser-Box Image:** 130px x 100px + 100px x 100px **Headline:** max. 30 characters incl. spaces **Teaser-Text:** max. 145 characters incl. spaces 4 motives max.	_	_	onvista.de
News-Feed	News will be imported via RSS-Feed/XML-Feed. We highly recommend RSS 2.0	_	_	onvista.de

	Format size in px (width x height)	size in kb	File formats	available on
Advertorial	Headline: max 100 characters incl. spaces Teaser: (optional): max 600 characters Text: no limitation Image: high resolution and landscape format (max 16:9) CTAs possible: please provide CTA-Text and URL Within text it is also possible to highlight and link words It is not possible to link images	_	jpg; png	onvista.de

Contact sales onvista media
sales@onvista.de

onvista mobile

	Format size in px (width x height)	size in kb	File formats	available on
Understitial	320 x 480 px	50	jpg, gif	onvista.de (Android App, iPhone App & mobile Webseite)
Content Ad	300 x 250 px (Android, iPhone, mobile Website)	30	bmp, jpg, gif, png	onvista.de (Android App, iPhone App & mobile Webseite)
Mobile Banner	2:1 300 x 150 px	30	bmp, jpg, gif, png	onvista.de (android app, iphone app & mobile webseite)
Premium-Box mobile	**Headline**: 55 characters incl. spaces **Teaser-Text:** 140 characters incl. spaces **Teaser-Box-Bild**: 160 x 120 **CTA-Link:** 15 characters	_	eps	onvista.de
Interactions	Headline: 34 Zcharacters incl. spaces Teaser-Text: characters incl. spaces (max. 3 lines) Button: 22 characters incl. spaces Image: quadratic size mind. 800x800 Pixel coloured or filled backround; no transparency	_	jpg	onvista Android App & iPhone App
Photo Ad Interactions	Headline: max. 40 characters incl. spaces Text: max. 160 characters incl. spaces Button-Text: max. 22 characters incl. spaces URL: Intern/Extern Tracking: UTM-parameter allowed, external Adserver not permitted Image: 16:9, minimum 1.200 x 675 pixel use as little light/white color as possible The image serves as background for headline and text	_	jpg	onvista Android App & iPhone App, onvista.de

Special forms of advertising

General points

Due to the asynchronous taggoing of our portals we request that you refrain from using the document write script. We will not otherwise be able to guarantee that the advertising material will be correctly implemented and displayed.

You should deliver the advertising material at least four days before campaign launch, in the case of Fireplace, 1 week in advance and in case of premium-box my onvista 14 days in advance.

Please supply the advertising material in an https-enabled format.

[1] HTML5

Size and maximal weights of the formats are according to the previous specifications.
If multiple files are used, the root file should always be named „index.html“.

HTML5 AD MEDIUM (implementing clickTags)

Click counting library
Ad the following script into the <head> area of the main ad medium file (index.html):
<script src="//ns.sascdn.com/diff/templates/js/banner/sas-clicktag-3.0.js"></script>

clickTag variables

```
<!DOCTYPE html>
<html>
<head>
<title>html5 multi clicktag</title>

<script src="//ns.sascdn.com/diff/templates/js/banner/sas-clicktag-3.0.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0"><style>
body{width:300px;height:600px;margin:0;padding:0;}
#main-container{width:300px;height:600px;cursor:pointer;}
#click-area1{background-color:red;height:150px;display:block;}
#click-area2{background-color:blue;height:150px;display:block;}
#click-area3{background-color:yellow;height:150px;display:block;}
#click-area4{background-color:black;height:150px;display:block;}
</style>
</head>
<body>

<div id="main-container">
<a id="click-area1"></a>
<a id="click-area2"></a>
<a id="click-area3"></a>
<a id="click-area4"></a>
</div>
<script type="text/javascript">
var clickArea1 = document.getElementById("click-area1");
clickArea1.onclick = function(){

window.open(clickTag0, "blank")

}
var clickArea2 = document.getElementById("click-area2");
clickArea2.onclick = function(){

window.open(clickTag1, "blank")

}
var clickArea3 = document.getElementById("click-area3");
clickArea3.onclick = function(){
```

```
window.open(clickTag2, "blank")

}
var clickArea4 = document.getElementById("click-area4");
clickArea4.onclick = function(){

window.open(clickTag3, "blank")

}
</script>
</body>
</html>
```

Single click URL

Declare the "ckickTag variable in the ad medium HTML file (index.html) and assign the click URL.

```
<script>var clickTag = "http://www.theclickthroughurl.com";
</script>
```

Multiple click URLs

If the ad medium contains several clickable elements, ze clickTags have to be listed (numerically): clickTag0, clickTag1, clickTag2 ...

```
<script>
var clickTag0 = "http://www.theclickthroughurl-1.com";
var clickTag1 = "http://www.theclickthroughurl-2.com";
var clickTag2 = "http://www.theclickthroughurl-3.com";
</script>

<script type="text/javascript">
var clickTag0 = "http://www.test1.de";
var clickTag1 = "http://www.test2.de";
var clickTag2 = "http://www.test3.de";
var clickTag3 = "http://www.test4.de";
</script>
```

Link assignment

If you assign a hyperlink to a clickable element on the ad (text, button, image etc.), you can choose one of the following options.

Option 1

```
<a id="clickArea"></a>

<script type="text/javascript">
var clickArea = document.getElementById("clickArea");
clickArea.onclick = function(){
window.open(clickTag, "blank");
}
</script>
```

Option 2

```
<a id="clickArea" target="_blank"></a>

<script type="text/javascript">
var clickArea = document.getElementById("clickArea");
clickArea.href = clickTag;
</script>
```

When using the 2nd option, you have to wait for the „click counting library“ to initialise. The „library“ first has to replace the Javascript clickTag variables before the ad can be rendered.
In order to be alerted when the initialisation is done, insert the "init"function (registered callback), which will notify you when the "click counting library“ has completed its task:

```
<script>
function init(){
/* this is a custom function which starts building the ad */
}
sas.clicktag.register(function(){
init();
});
</script>
```

Examples of use:

Example 1 : Simple clickTag using the „window.open“ method

```
<!DOCTYPE html>
<html>
<head>
<title>html5 single clicktag</title>

<script src="//ns.sascdn.com/diff/templates/js/banner/sas-clicktag-3.0.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0">
<style>
body{width:300px;height:600px;margin:0;padding:0;}
#main-container{width:300px;height:600px;cursor:pointer;}
#click-area1{background-color:red;height:600px;display:block;}
#click-area2{background-color:blue;height:150px;display:block;}
#click-area3{background-color:yellow;height:150px;display:block;}
#click-area4{background-color:black;height:150px;display:block;}
</style>
</head>
<body>

<script type="text/javascript">
var clickTag = "http://www.test.de";
</script>

<div id="main-container">
<a id="click-area1"></a>
</div>
<script type="text/javascript">
var clickArea1 = document.getElementById("click-area1");
clickArea1.onclick = function(){

window.open(clickTag, "blank")

}
</script>
</body>
</html>
```

Example 2: Multiple ClickTags using the “window.open” method

```
<!DOCTYPE html>
<html>
<head>
<title>html5 multi clicktag</title>

<script src="//ns.sascdn.com/diff/templates/js/banner/sas-clicktag-3.0.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0">
<style>
body{width:300px;height:600px;margin:0;padding:0;}
#main-container{width:300px;height:600px;cursor:pointer;}
#click-area1{background-color:red;height:150px;display:block;}
#click-area2{background-color:blue;height:150px;display:block;}
#click-area3{background-color:yellow;height:150px;display:block;}
#click-area4{background-color:black;height:150px;display:block;}
</style>
</head>
<body>

<script type="text/javascript">
var clickTag0 = "http://www.test1.de";
var clickTag1 = "http://www.test2.de ";
var clickTag2 = "http://www.test3.de ";
var clickTag3 = "http://www.test4.de ";
</script>

<div id="main-container">
<a id="click-area1"></a>
<a id="click-area2"></a>
<a id="click-area3"></a>
<a id="click-area4"></a>
</div>
<script type="text/javascript">
var clickArea1 = document.getElementById("click-area1");
clickArea1.onclick = function(){

window.open(clickTag0, "blank")

}
var clickArea2 = document.getElementById("click-area2");
clickArea2.onclick = function(){

window.open(clickTag1, "blank")

}
var clickArea3 = document.getElementById("click-area3");
clickArea3.onclick = function(){

window.open(clickTag2, "blank")

}
var clickArea4 = document.getElementById("click-area4");
clickArea4.onclick = function(){

window.open(clickTag3, "blank")
}
</script>
</body>
</html>
```

Example 3 : Simple clickTag using the “registered callback” function

```
<!DOCTYPE html>
<html>
<head>
<title>html5 single clicktag with callback test</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0">

<script src="//ns.sascdn.com/diff/templates/js/banner/sas-clicktag-3.0.js"></script>

<style>
body{width:300px;height:600px;margin:0;padding:0;}
#main-container{width:300px;height:600px;cursor:pointer;}
#click-area1{background-color:red;height:600px;display:block;}
#click-area2{background-color:blue;height:150px;display:block;}
#click-area3{background-color:yellow;height:150px;display:block;}
#click-area4{background-color:black;height:150px;display:block;}
</style>
</head>
<body>

<script type="text/javascript">
var clickTag = "http://www.test.de";
</script>

<div id="main-container">
<a id="click-area1" target="_blank"></a>
</div>


<script>
function customerFunc(){
var clickArea1 = document.getElementById("click-area1");
clickArea1.href = clickTag;
}
sas.clicktag.register(function(){
customerFunc();
});
</script>

</body>
</html>
```

[2] Sitebar

The Sitebar is a dynamic ad format, which follows the screen resolution. So the ad has to be created as responsive format.

[3] Homepage Exclusive and Area Bookings

Materials should always be delivered physically. It is also possible to integrate them as redirect or iFrame. Server utilisation must be observed.

The individual elements may not be larger than 80 kb.
Mx. 3 motives are placed in rotation.

[4] Standalone Newsletter onvista.de

Format: HTML
You should deliver the newsletter 5 days in advance.
Size max. 60kb (incl. All files)
Subject max. 70 signs

All graphics shown in the newsletter must be delivered physically. Referencing onto an external customer server is not permitted.

For optimal representation in all standard mail clients, please observe the following:

Use a table layout.
Use only static content (no JavaScript, no Flash).
No forms.
No anchor links as these cannot be used globally.
No external css files, no css definitions in the header >> only inline styles.
No justified text (e.g. align=„justify“).
No background images (e.g. as a colour gradient or as a background in a table)
No DIVs.
Give preference to HTML Tags before style attributes (e.g. <p align="right"> statt <p style="text-align:right;">).
Script formatting should generally be determined for all paragraphs with <font> tag, not with an inline style (Outlook 2007 problems).
Images within tables (e.g. frames) may be compressed at most but never stretched (Outlook 2007 problems).
Avoid URLs in the linking text / anchor text, in particular http://--TargetUrl--as some mail clients see these as a phishing attempt.

A final control will be carried out by our department. The newsletter supplied by you will be tested for the correct representation.

With the above mentioned specifications, a clean display is guaranteed in the standard mail programmes.
If these specifications are not complied with, display errors may occur (especially in Lotus Notes and Outlook 2007).