

IAB Switzerland Anlieferung

Stand: 10. März 2010

Flash Spezifikation

Bei Flash Werbemitteln ist die Klickzählung nicht automatisch durch den Adserver steuer- und implementierbar. Deshalb ist bei der Erstellung ein spezieller ActionScript Code zu implementieren, um die Zählung auf Adservern zu gewährleisten.

Der Adserver muss dem Flash Werbemittel seine Klick-URL als Parameter übergeben, so dass eine statistische Auswertung möglich wird. Dieser Parameter, „clickTAG“ (Gross-/Kleinschreibweise beachten!), wird vom Flash-Code ausgelesen und die so übergebene Klick-URL beim Klick in einem neuen Fenster geöffnet. Die Zielseiten werden also nie im Werbemittel selber definiert, sondern vom Adserver übergeben. So ist auch eine Ziel-URL Änderung während der Kampagnenlaufzeit ohne Werbemittelanpassung, sowie die Auswertung der Klicks durch den Adserver möglich.

Es sollen wenn immer möglich keine externen Ressourcen (z.B. vom Firmen-Webserver) geladen werden. Falls dies trotzdem notwendig ist, muss die crossdomain-Policy beachtet und gesetzt werden, so dass der Flash-Film überhaupt auf die externen Ressourcen zugreifen kann.

Es ist generell empfehlenswert nicht die aktuellste Flash Version einzusetzen. Damit wird die bestmögliche Verbreitung Ihres Sujets ermöglicht. Aktuell wird Flash Version 9 als höchste Version empfohlen.

Klick-Zählung

Über ActionScript 2 (Standard bei Flash Version 8 und früher) wird der Klick-Tag wie folgt implementiert:

```
/**
 * Code in Button
 */
on (release) {
    getURL( _root.clickTAG, "_blank" );
}
```

Über ActionScript 3 (Standard bei Flash 9 und folgenden) wird der Klick-Tag wie folgt implementiert:

```
/**
 * searches for a clickTAG, case-insensitive search
 */
function clickTagGet():String {
    for (var key:String in LoaderInfo(this.root.loaderInfo).parameters) {
        if (key.toLowerCase()=='clicktag') {
            return LoaderInfo(this.root.loaderInfo).parameters[key];
        }
    }
    return null;
}
/**
 * searches for a clickTARGET, case-insensitive search
 */
function clickTargetGet():String {
    for (var key:String in LoaderInfo(this.root.loaderInfo).parameters) {
        if (key.toLowerCase()=='clicktarget') {
```

```

        return LoaderInfo(this.root.loaderInfo).parameters[key];
    }
}
return null;
}
/**
 * Opens the clickTAG and prevents to be blocked as a popup
 */
function clickTagOpen(url:String=null, target:String=null):void {
    if (!target) {
        target=clickTargetGet();
        if (!target) {
            target='_blank';// default: to new window
        }
    }
    if (!url) {
        url=clickTagGet();
    }
    if (url) {
        var req:URLRequest=new URLRequest(url);
        if (!ExternalInterface.available) {
            navigateToURL(req, target);
        } else {
            var strUserAgent:String=String(ExternalInterface.call("function() { return navigator.userAgent;
            })).toLowerCase();
            if (strUserAgent.indexOf("firefox") != -1 || (strUserAgent.indexOf("msie") != -1 &&
            uint(strUserAgent.substr(strUserAgent.indexOf("msie") + 5, 3)) >= 6)) {
                ExternalInterface.call("window.open", req.url, target);
            } else {
                navigateToURL(req, target);
            }
        }
    }
    } else {
        trace('clickTAG was not found.');
```

Mehrere Ziel-URL's:

Sobald das Werbemittel auf verschiedene Ziel-Seiten verlinkt, ist es u.U. gewünscht, diese getrennt zu messen. Hierzu werden vom Adserver jeweils mehrere Klick-URL's über die Parameter clickTAGX übergeben (X steht für eine fortlaufende Link-Zahl).

Bei der Anlieferung des Werbemittels ist hierzu jeweils anzugeben, welcher Parameter auf welche Ziel-URL verweisen soll.

Beispiel:

„clickTAG1“ entspricht http://www.zielseite.ch/unterseite_1/
 „clickTAG2“ entspricht http://www.zielseite.ch/unterseite_2/
 „clickTAG3“ entspricht http://www.zielseite.ch/unterseite_3.html
 u.s.w

ActionScript 2 Beispiel:

```

/**
 * Code in Button 1
 */
on (release) {
    getURL( _root.clickTAG1, "_blank" );
}
```

```

/**
 * Code in Button 2
 */
on (release) {
    getURL( _root.clickTAG2, "_blank" );
}

```

ActionScript 3 Beispiel:

```

/**
 * searches for a clickTAG, case-insensitive search
 */
function clickTagGet(clickNumber:String):String {
    for (var key:String in LoaderInfo(this.root.loaderInfo).parameters) {
        if (key.toLowerCase()=='clicktag' + clickNumber) {
            return LoaderInfo(this.root.loaderInfo).parameters[key];
        }
    }
    return null;
}
/**
 * searches for a clickTARGET, case-insensitive search
 */
function clickTargetGet(clickNumber:String):String {
    for (var key:String in LoaderInfo(this.root.loaderInfo).parameters) {
        if (key.toLowerCase()=='clicktarget' + clickNumber) {
            return LoaderInfo(this.root.loaderInfo).parameters[key];
        }
    }
    return null;
}
/**
 * Opens the clickTAG and prevents to be blocked as a popup
 */
function clickTagOpen(clickNumber:String, url:String=null, target:String=null):void {
    if (!target) {
        target=clickTargetGet(clickNumber);
        if (!target) {
            target='_blank';// default: to new window
        }
    }
    if (!url) {
        url=clickTagGet(clickNumber);
    }
    if (url) {
        var req:URLRequest=new URLRequest(url);
        if (!ExternalInterface.available) {
            navigateToURL(req, target);
        } else {
            var strUserAgent:String=String(ExternalInterface.call("function() { return navigator.userAgent;
}").toLowerCase());
            if (strUserAgent.indexOf("firefox") != -1 || (strUserAgent.indexOf("msie") != -1 &&
uint(strUserAgent.substr(strUserAgent.indexOf("msie") + 5, 3)) >= 6)) {
                ExternalInterface.call("window.open", req.url, target);
            } else {
                navigateToURL(req, target);
            }
        }
    }
    } else {
        trace('clickTAG' + clickNumber + ' was not found.');
```

```

}
button1_instanzname.addEventListener(MouseEvent.CLICK, function(e:MouseEvent) { clickTagOpen('1'); });
button2_instanzname.addEventListener(MouseEvent.CLICK, function(e:MouseEvent) { clickTagOpen('2'); });

```

Layer (Floating) und Expandable Ads

Close-Button

Alle Layer- und Expandable Werbemittel (wenn diese von Anfang an ausgeklappt sind) müssen über einen funktionierenden Close-Button verfügen. Dieser muss jederzeit gut sichtbar (besonders auf weissem Hintergrund) und mindestens 15x15 Pixel gross sein. Sobald der Close-Button gedrückt wird, muss eine vom Adserver vorgegebene Javascript-Methode oder fscommand aufgerufen werden, so dass spezifische Close-Aktionen vom Adserver ausgeführt werden können (wie zum Beispiel das umliegende DIV-Element verstecken). IAB Schweiz empfiehlt hier die Verwendung von „*javascript:iab_close()*“ oder *fscommand(,iab_close)* im Flash-Werbemittel. Die Javascript-Funktion bzw. fscommand Handler muss vom Adserver entsprechend definiert sein.

Automatisches Schliessen

Layer Ads (Floating Ads) müssen sich nach einer Laufzeit von 7 Sekunden automatisch selber schliessen. Gleiches gilt für Expandable-Banner die im Anfangszustand bereits ausgeklappt sind.

Werbemittel mit Sound

Werbemittel mit Sound müssen im „SoundOff“-Modus starten. Der Sound soll über einen Button vom Benutzer aktiviert/deaktiviert werden können oder darf frühestens starten, wenn die Maus über dem Werbemittel positioniert wird (MouseOver) bzw. wieder enden, sobald die Maus das Werbemittel verlässt (MouseOut).

Klickbereiche und Transparenz

Normalerweise werden Layer- und Expandable Werbemittel über *wmode="transparent"* ausgeliefert, so dass der Website Content unter dem Flash-Banner sichtbar wird. Hierbei muss sichergestellt werden, dass nur die sichtbaren Elemente des Flash-Banners klickbar sind. Weiter ist zu beachten, dass die Hintergrundfarbe transparent dargestellt wird (falls ein Bereich mit Hintergrundfarbe trotzdem dargestellt werden soll, muss ein Objekt mit entsprechender Farbe in Flash definiert werden).

Öffnen und Schliessen von Werbemitteln

Bei Expandable-Banner und Banner mit Eventtracking sind jeweils spezielle, vom Adserver definierte Javascript-Methoden oder fscommands aufzurufen.

Für die Expand-Aktion empfiehlt IAB Switzerland die Verwendung des Aufrufs: „*iab_show*“ (entweder über gleichnamige Javascript-Funktion oder fscommand-Parameter) und für die Collapse-Aktion entsprechend: „*iab_hide*“. Diese sind notwendig, damit das umliegende DIV vergrössert/verkleinert werden kann (nur so ist der Content der Website im Hintergrund des vergrösserten Werbemittels weiterhin klickbar, nachdem das Werbemittel sich wieder auf die ursprüngliche Grösse reduziert hat).

Rich-Media Spezifikation

Im Folgenden werden grundsätzliche Richtlinien beschrieben, die bei der Produktion von Rich-Media Werbemitteln beachtet werden sollen.

Ein auf HTML/JS basierendes Rich-Media Werbemittel besteht in erster Linie aus einem HTML-Code, welcher andere Ressourcen (wie Bilder, CSS, Flash) nachlädt. Die zusätzlichen Ressourcen liegen – falls vom Adserver nicht anders vorgegeben – im selben Verzeichnis und werden vom Adserver meistens durch ein Platzhalter oder den Dateinamen repräsentiert. Bei der Anlieferung wird empfohlen, die Dateinamen ohne Angabe eines Pfades/Domain im Quelltext zu verwenden.

Richten Sie sich bezüglich Klick-Zählung bei Rich-Media Werbemitteln an die Spezifikation des jeweiligen Adservers.

Bei der Erstellung von Rich-Media Werbemitteln ist weiter folgendes zu beachten:

- HTML-Tags, die nach dem HTML-Standard ausserhalb des <body>-Bereichs eingesetzt werden, dürfen nicht verwendet werden. Dies sind unter anderem folgende:
 - <html>
 - <head>
 - <title>
 - <meta>
 - <body>
- Initialisierung (keine Ereignisse wie onload): Javascript-Teile des Rich-Media Banners müssen ohne die Einbindung in Ereignisse der restlichen Website lauffähig sein. Die Einbindung in die Website erfolgt meistens über einen <script>-Tag äquivalenten Aufruf.
- Es sollten wenn immer möglich keine externen Ressourcen (z.B. vom Firmen-Webserver) geladen werden.
- Alle Dateien sollen in einem Verzeichnis und als ZIP-Datei angeliefert werden
- Das Werbemittel soll bei der Erstellung auf mehreren Browsern und Plattformen getestet werden
- Falls es sich um ein Expandable-Werbemittel handelt:
Für die Expand-Aktion empfiehlt IAB Switzerland die Verwendung des Aufrufs:
„`javascript:iab_show()`;“ und für die Collapse-Aktion entsprechend:
„`javascript:iab_hide()`;“
- Falls es sich um ein Layer Ad mit Close-Button handelt, ist der Klick auf den Close-Button dem Adserver über den Aufruf „`javascript:iab_close()`;“ mitzuteilen