API QRG

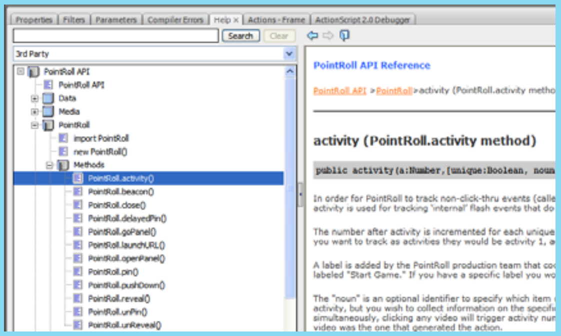
Legend

Items in blue include general API information.

Items in purple apply to banners only.

Items in green apply to panels only.

Items in yellow apply to banners and panels.



PointRoll API Reference

PointRoll API > PointRoll > activity (PointRoll.activity metho

activity (PointRoll.activity method)

```
public activity(a:Number,[unique:Boolean, noun
```

In order for PointRoll to track non-click-thru events (calle
activity is used for tracking 'internal' flash events that do

The number after activity is incremented for each unique
you want to track as activities they would be activity 1, a

A label is added by the PointRoll production team that co
labeled "Start Game." If you have a specific label you wo

The "noun" is an optional identifier to specify which item
activity, but you wish to collect information on the specifi
simultaneously, clicking any video will trigger activity nur
video was the one that generated the action.

Code Definitions

- **PointRoll Object** - Used to access the core functionality of PointRoll's features, this object requires the PointRoll Flash API to run. You need to declare the PointRoll object in the first frame of your ActionScript layer.
- **Activity Tracking Code** - Use this code for tracking internal Flash events that do not spawn a new window; e.g., a start button on a game or scrollbars in the SWF.
- **clickTag Code** - This code is for tracking click-throughs; i.e., the user clicks somewhere in the ad and launches a new window.
- **Close Command** - This command is used to close out a panel either as a button in Flash or on the last frame of an auto-display panel. This provides the seamless effect of going from a BadBoy floating ad to a FatBoy leave-behind.
- **Pin Command** - This command keeps the panel open once a user clicks on a form input field. For example, if the panel contained a form and the user started to fill it out but accidentally rolled off of the panel, the panel would then close. This code keeps the panel open once the user clicks on an input field. Note: If the pin command is used, then a corresponding close button must be prominent.
- **Absolute Path for External Files** - Code may be used to pull in the absolute path dynamically for external files such as sub-SWFs, JPGs, PNGs, and MP3s.
- **Flash Action** - This may be placed in a banner that does not use the PointRoll image map tool to create the hotspots and track click-throughs in the banner. This code expands the panels and tracks click-throughs via the Flash banner rather than the transparent GIF overlay.
- **goPanel** - This method allows you to go from one panel to another.

Creating the PointRoll Object

When working with a PointRoll creative utilizing the API, it is imperative to create a PointRoll object to handle the API functions. In frame 1 of your ActionScript layer*, enter the following in the Actions window:

```
import PointRollAPI.PointRoll
var myPR:PointRoll = new PointRoll(this);
```

**If you don't have an ActionScript layer, create one.*

You can name the PointRoll object any name you wish; “myPR” is the name of the object for the purposes of this document. You must have a separate PointRoll object for each banner and panel FLA file if you wish to use PointRoll API functions. Keep in mind of the scope of your PointRoll object when referring to it from within movie clips or buttons.

API QRG

Activity Tracking Code

As of PointRoll API version 4.4.0, activities can be tracked from banners as well as panels. There are three versions of this code. One tracks events on release/press so that if the user continues to click or interact with the tracked event, the code continues to fire off. The second is for “on rollOver” events and fires off the activity tracking only once, preventing useless tracking metrics. The third allows the association of a “noun” with the activity, which is useful for campaigns using dynamic data.

The number after “activity” must be incremented for each unique tracked event. It starts out at 1 for each panel. The on release/press events example below is tracking activity 2.

```
on (release) {
  myPR.activity(2);
}
```

For “on rollover” events, it is important to use “true” as the second attribute. This flags the activity as unique so the variable doesn't reset and fire off again continuously. This example is tracking activity 3.

```
on (rollOver) {
  //...your ActionScript for the event goes here...
  myPR.activity(3, true);
}
```

For campaigns utilizing dynamic data feeds, the same activity numbers are used for many different objects. In order to clarify the activity reporting, you may specify the name of the object in the activity call.

```
on(rollover){
  //...your ActionScript for the event goes here...
  myPR.activity(3, false, myProductName);
}
```

In order to utilize the “noun” parameter, you must pass all three values (activity number, false, product name).

Click Tag Code

It is important to use “_root” in the following code:

```
on (release) {
myPR.launchURL (_root.clickTag1);
}
```

The number after “clickTag” is incremented for each additional click, so the second click-through would be:

```
on (release) {
myPR.launchURL (_root.clickTag2);
}
```

Close Command

To add a clickable close event in Flash, add this exact action to a button object:

```
on (release) {
  myPR.close();
}
```

To close the panel at the end of an animation or movie piece (especially useful for PointRoll BadBoys and TowelBoys), add this action to the appropriate frame:

```
myPR.close();
```

API QRG

Pin Command

For pinning to occur upon clicking a button with the API, add this script to the action for the event:

```
on (release) {
    myPR.pin();
}
```

Pinning for a Flash 6 + Form

Pinning a panel when a user clicks on a form eliminates the possibility of the panel closing accidentally upon rollOut, forcing the user to enter the information in the form all over again. In the example below, “lname_txt,” “fname_txt” and “address_txt” are the instance names for the fields that should trigger pinning. This should be on a frame that spans the timeline.

```
lname_txt.onSetFocus = fname_txt.onSetFocus = address_txt.onSetFocus = function ()
{
myPR.pin();
}
```

Absolute Path Code

When externally loading sub-SWFs, images or music files, it is imperative to place the absolute path code before entering the URL of the externally loaded file. This code is a string that you can concatenate to the location of the file. On the event that is used to load, just reference that file with this code:

```
loadMovie(myPR.absolutePath +"my_file.swf", mySUB);
```

“mySUB” is the instance name of the movie clip or object you are loading the file into. This should be changed to the instance name you define in your FLA.

The same effect can also be achieved this way:

```
mySUB.loadMovie(myPR.absolutePath + "my_file.swf");
```

You may need an RMS engineer to assist you in uploading more than nine external files. Otherwise, you can upload them along with your panel parent SWF in AdPortal. External files need to be in the same directory as the parent SWF. They cannot be in subdirectories.

If you upload your file and the absolute path code is not pulling your external files, please contact RMS. Also, unlike our normal panels, you need to clear your cache when previewing panels with external files if you make any changes.

goPanel

Sometimes it is necessary to have the expanded panels overlap the main banner. However, this covers up the rollover hotspots on the main creative, making the other panels inaccessible. To remedy this, you can add API code to the panel to launch subsequent panels. This can be done on a Flash event, such as “on release” or “on rollOver.”

To go from panel to panel, you need to use the following:

```
on (release) {
    myPR.goPanel(1);
}
```

Using “prGoPanel” tracks the panel opening as an interaction in the reporting (just as opening any panel). It is not tracked as an activity.

API QRG

Flash Action Banners

Expanding the Panel(s)
This is the code for the buttons in the Flash banner that expands the panel(s).

Place this code on the actual button(s):

```
on (rollOver) {
myPR.openPanel(1);
}
```

The 1 in the example above is changed to the number of the panel you want to open. The code for panel 2 is shown below.

```
on (rollOver) {
myPR.openPanel(2);
}
```

Continue to increment the number for each additional panel.

Tracking Banner Click-throughs

```
on (release){
    if (typeof(_root.clickTag1) == "undefined")
    {
        var dest = "http://www.LandingURL1.com";
        myPR.launchURL(dest);
    }
    else
    {
        myPR.launchURL(_root.clickTag1);
    }
}
```

For additional clicks in the banner, just increment the "clickTag1" to "clickTag2" and add the new URL in the destination variable.

```
on (release){
    if (typeof(_root.clickTag2) == "undefined")
    {
    var dest = "http://www.LandingURL2.com";
        myPR.getURL(dest);
    }
 else
    {
        myPR.getURL(_root.clickTag2);
    }
}
```

Each new click uses the same code, with the exception of incrementing the clickTag and changing the URL.

PointRoll Checklist

Make sure that you've checked the following items against the appropriate publisher specs.

Banner to Spec
- Weight
- 18 fps or lower
- Animation restriction
- Flash Action code (when applicable)

Panel(s) to Spec
- Weight
- Width/height
- 18 fps or lower
- clickTag tracking
- Activity tracking
- Pinning for forms
- Absolute path for sub-SWFs

Default Banner
- Weight
- Animation restriction on animated GIF

Digital Video
- MOV, AVI or Digital Beta

Widget Menu

Invoking the PrWidgetControl class calls the Clearspring menu within the ad unit so that users can snag the widget.

```
//import the widget control class
import PointRollAPI.util.widgets.PrWidgetControl;

//create a widget control object
var prWidget:PrWidgetControl= new PrWidgetControl();

//call the showMenu method, within the onRelease function of any button
grabIt_btn.onRelease = function(){
prWidget.showMenu();
}
```

The PrWidgetControl class automatically looks for the widget ID in a variable called _root.prWID. This value must be set as a panel variable within AdPortal, and should never be hard-coded into the creative (similar to the use of _root.prVidMethod for streaming or progressive video delivery).

Questions about this process? Please email richmediasupport@pointroll.com.