



INSTRUMENT LABS

SOCIAL GRID DOCUMENTATION

WELCOME!

Social Grid is a media visualization project from the Labs team at Instrument. We created the project to display event-based media (Twitter, Instagram, and event-specific content) in a dynamic and visually interesting way. We've had a blast with it at our own events, and we think it's about time to share it with the community.

We added some fun customization features for this release, as well as the ability to display video content. So right out of the box, you can use Social Grid to display Twitter, Instagram, and your own images and video. The project is also fully open source, so you are also welcome to develop your own features, if that's what you're into.

PACKAGE COMPONENTS

There are three distinct parts to this package. The essential component is the **Social Grid Display App**, and the **Backend Moderation Tool** and **Photobooth iPad App** can be used to add to the experience. Here's a rundown of the components:

SOCIAL GRID DISPLAY APP

The part of the project that displays and manages flow of content is a desktop Air Application written in Actionscript 3.0. When configured correctly (see below), the Social Grid Display App can run independently of the other components.

BACKEND MODERATION TOOL

To make the integration of social media a bit more robust, we created an optional backend tool that runs on a server. It can be configured to aggregate social media content by hashtag and used to moderate the content that appears in the Social Grid Display App.

PHOTOBOOTH IPAD APP

During our 2012 Halloween party, we set up two photo booths that sent photos to be displayed in the Social Grid Display App's content stream. The Photobooth iPad App we made is included in this package, and can be used to send photos to the Social Grid Display App.

INSTALLATION

This section describes the basic system environment and actions needed to get each component of the package installed and running.

SOCIAL GRID DISPLAY APP

The package includes a build folder, with an Adobe Air application installer, **SocialGrid.air**. You may need to install Adobe Air on your system in order to run the installer. Simply open the installer, and follow the prompts to install the application. To run it, navigate to the installation location and open the **SocialGrid** application.

When the app is running, you can press the **F** key at any time to toggle fullscreen on and off, and the **M** key to toggle mouse visibility.

The application's config.xml file is located inside the installed application package, inside a folder called resources. Depending on your operating system, you may need to view the application package contents in order to access the configuration file. See below for configuration details.

BACKEND MODERATION TOOL

The backend tool runs using **Node.js**. For information on Node.js, please visit <http://nodejs.org>. Once you have Node.js installed on your server, you'll also need the following packages:

```
npm install bindings
npm install contextify
npm install mysql
npm install node-static
npm install ntwitter
npm install oauth
```

We also suggest installing forever and using it, because it's just plain awesome:

```
npm install forever
```

You'll also want to set up a MySQL database. Here's the schema for the tables:

(see next page)

```
CREATE TABLE `social_grid_data` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `type` varchar(30) DEFAULT NULL,
  `text` varchar(160) DEFAULT NULL,
  `image` varchar(500) DEFAULT NULL,
  `approved` int(11) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `social_id` varchar(30) NOT NULL,
  `name` varchar(100) DEFAULT NULL,
  `updated_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`id`),
  UNIQUE KEY `social_id` (`social_id`)
)
```

```
CREATE TABLE `social_grid_shown` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `parent_id` int(11) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `par_ind` (`parent_id`),
  CONSTRAINT `social_grid_shown_ibfk_1` FOREIGN KEY (`parent_id`)
  REFERENCES `social_grid_data` (`id`) ON DELETE CASCADE
)
```

Once the system is up and running, you can use the following pages in a modern web browser:

/live-manage	Live streaming updates from Instagram and Twitter appear on this page with a 10 second countdown and an accept or reject button. All items are automatically approved unless rejected on this page.
/all	This page displays all of the recent items matching your hashtags. You can remove unwanted items from the stream here.
/shown	This page displays all of the items recently shown on the display. You can remove unwanted items from the display here.

PHOTOBOOTH IPAD APP

The Photobooth iPad App is not available in the app store, so it must be compiled and deployed directly to the device via **XCode**. See Development (below) for more details.

A web server is also needed to handle image uploaded from the photobooth. **XAMPP** is a simple web server based on Apache that is available for Mac, Windows, and Linux (<http://www.apachefriends.org/en/xampp.html>) and works well for this purpose. Once installed, the included sample PHP file can be used to process uploaded images.

CONFIGURATION

The components of this package can be customized via configuration files and/or application settings. This section lists the options, with example values as needed to clarify.

SOCIAL GRID DISPLAY APP

Found in the installed app's resources folder, the **config.xml** file can be used to customize colors and other program settings. Here's a rundown of the options.

loadingBackgroundColor	The color of the loading screen background. <code>#ffffff</code>
loadingCircleColor	The color of the loading circle.
loadingIconColor	The color of the loading icon.
loadingTextColor	The color of the loading text.
twitterTextColor	The color of Twitter content text.
twitterHighlightColor	The color of highlighted Twitter content text (usernames, hashtags, etc.).
twitterBackgroundColor	The color of Twitter content background tiles.
largeTwitterTextColor	The color of large Twitter content text.
largeTwitterHighlightColor	The color of large highlighted Twitter content text.
largeTwitterBackgroundColor	The color of large Twitter content background tiles.
instagramTextColor	The color of Instagram content text.
instagramHighlightColor	The color of Instagram Twitter content text.
pullSocialDirectly	Whether or not to pull social media content directly from their respective services. If true, backend moderation tool isn't used. <code>true/false</code>
twitterHashtags	A comma-separated list of Twitter hashtags to use when pulling in social media directly (pullSocialDirectly must be set to true). <code>instrumentoutpost, examplehashtag</code>
instagramHashtags	A comma-separated list of Instagram hashtags to use when pulling in social media directly (pullSocialDirectly must be set to true). <code>instrumentoutpost, examplehashtag</code>

backendRoot	The base url of Backend Moderation Tool installation. To use the backend, pullSocialDirectly must be set to false. <code>http://example.com/example</code>
photoboothFolderPath	A relative path from the application to the folder which contains photobooth images. <code>resources/photobooth</code>

BACKEND MODERATION TOOL

You will find a config file at `/credentials/env.js`.

host	This is the host name your moderation tool will be running on.
mysql	Put your connection info here. Check the mysql Node package manual if you are having trouble connecting.
twitter	Get your app credentials from http://dev.twitter.com .
instagram	Get your app credentials http://api.instagram.com .
tags	A comma-separated list of Twitter and Instagram hashtags to use when pulling in social media.
username	If you want your moderation to be protected, enter a username here.
password	If you want your moderation to be protected, enter a password here.
ignoreSwears	Whether or not to block social media content that contains inappropriate language.

PHOTOBOOTH IPAD APP

The iPad app has two configuration options, these options are found in the standard iOS Settings app under the PhotoBooth section.

Post URL	The URL to which photos will be uploaded via a standard HTTP post. <code>http://example.com/example</code>
Source Name	An optional value that will also be included in the HTTP post and can be used to identify the source of the upload (in the even that there are multiple photo booth instances in use simultaneously). <code>example_name</code>

DEVELOPMENT

Because we'd love to see what you can do with this code, it's all included in this open-source package. That said, it's time for our disclaimer. We made these applications with the main goal of simply getting them working for our events. While some of the code is commented, a lot of it isn't. We want you to go for it, but realize that you'll have to take it from here!

SOCIAL GRID DISPLAY APP

We developed the Social Grid Display App as an Actionscript project in **Adobe Flash Builder 4.6**, and we'd recommend using the same development environment. Just use **File > Import Flash Builder Project**, and you'll be ready to pick up where we left off.

BACKEND MODERATION TOOL

In order to continue development of the Backend Moderation Tool, you'll simply need to have access to the MySQL database that you setup during installation (see above). Just open the project in your favorite code/text editor, and you're ready to jam.

PHOTOBOOTH APP

The photobooth app is not publicly available in the app store, it must be compiled and deployed via XCode.

Requirements for building the app:

- 1) XCode with iOS SDK 5.1 or later
- 2) Cinder version 0.8.4 (www.libcinder.org)

All graphical assets are PNG images located in the "assets" folder. Changing these assets may also require changing the code related to the visual layout if the image dimensions are changed. All rendering code can be found in the main C++ file "src/PhotoBoothApp.cpp" which includes inline comments.

A web server of some kind must be available to handle the photo uploads, a simple PHP script has been included with the source for this project but any web server that can process a multi-part HTTP post will work. The photobooth app must be configured to point to the web server URL or the app may timeout and crash when attempting to upload a photo. The post will contain the following:

image

The jpeg image, the filename will always be "image.jpg" and should be renamed to something unique on the server side.

source

The value of the "Source Name" setting, this can be ignored if not needed.