EXHIBIT 8

Using Activity Streams Last modified Tuesday, April 21, 2009 at 5:30pm by Pete Bratac

Using Activity Streams to Read User Data

Facebook syndicates users' streams, including content from both the News Feed and the Wall. Your applications and sites can read and display the stream in your application or site. One way you can do this is through the open Activity Streams standard, which gets sent to you as an Atom feed.

For information about streams, read Using the Stream API.

Before you can get stream content, the user must have granted your application or site **read_stream** extended permission. See Prompting for Permission to Access Streams for details on the permission.

Once the user has granted the permission, you can get that user's stream from the following URL:

http://www.facebook.com/activitystreams/feed.php

Reading the User's Stream

To read the user's stream with the intention of displaying it on your application or site, you would construct the URL with the following parameters:

http://www.facebook.com/activitystreams/feed.php?source_id=<user-id>&app_id=<yourApplicationId>&session_key=<session_key>&sig=<checksum-slash-signatutre>&v=0.7&read&updated_time=<UnixTime>

The GET parameters you need to append to the URL are:

- source_id: The user ID of the user whose stream you are accessing.
- app_id: Your application's ID.
- session_key: The active session for the current user.
- sig: A signature verifying that the request is coming from your application.
- read: A parameter whose presence dictates that your application is pulling the user's stream, provided your application has permission to do so.
- updated_time: (optional) If you include this parameter with your request, Facebook will only include posts created after that specified time. Specify this parameter as Unix time.

You must pass these GET parameters with the URL. Facebook ignores any other parameters you pass.

The sig parameter in particular is necessary so Facebook can easily verify the source of the HTTP request and trust that the user's stream is truly being sent to a known Facebook Connect site or Platform application.

The signature is an MD5 hash of a string involving the application ID, the source ID, the session key, and the application secret (or the session secret if you have a desktop application or you don't want to pass the application secret). You can get a session secret by calling auth.getSession and setting `generate_session_secret` to true.

For example, if the user ID is 44444, your application ID is 12345, your active session key is BBBBB, and your application secret is WWWWW, then the signature needs to be the MD5 hash of:

```
app_id=12345session_key=BBBBBsource_id=44444WWWWW
```

The MD5 hash of app_id=12345session_key=BBBBBsource_id=44444WWWWW is 9d27f75779c650e5769d64f28b7b7e4e, so the actual URL you would pass in this case would be:

http://www.facebook.com/activitystreams/feed.php?source_id=44444&app_id=12345&session_key=BBBBBsig=9d27f75779c650e5769d64f28b7b7e4e&v=0.7&read

Desktop applications should compute signatures precisely the same way, except that the session secret should be used instead of the application secret.

Response Codes

Like most HTTP responses, Facebook Activity Streams responses include a response header, which always includes a traditional response code and a short response message. The supported response codes include:

- 200 Code provided whenever the Facebook servers were able to accommodate the request and provide a response.
- 304 Code provided whenever the request header included If-Modified-Since and no new posts have been generated since the specified time.
  '''Note:''' Code 304 will never be returned if If-Modified-Since isn't included in the request header.
- 401 Code provided whenever the URL omits one or more of the required parameters.
- 403 Code provided whenever the URL is syntactically valid, but the user hasn't granted the required extended permission.
- 404 Code provided whenever the URL is syntactically valid, but the signature is incorrect, or the session key is invalid.

What Data Applications Can Republish

Once you pull the user's stream, you can republish this content on your site or in your application. The types of user posts that your application can republish depends upon the privacy setting the user applied to each specific piece of content. The types of content currently available include:

- Status updates and shared items, only if privacy is set to friends, friends of friends, networks, or everyone
- Videos, only if privacy is set to everyone
- Photos, only if privacy is set to everyone
- Links, only if privacy is set to everyone
- Notes, only if privacy is set to everyone

Any posts you publish using this content are built on the Atom feed framework. See some examples below.

Note: At this time, the Activity Stream feed contains posts that were generated by user-created content, but not by applications.

Sample Activity Stream Entries

The following entry represents a post generated because a user, Snapshot Smith, uploaded a single photo to her Wall. Most of the content within the <entry> tag is traditional Atom, and is the very material that all popular Atom readers like Google Reader and NewzCrawler use to render the content. The <activity:verb> and <activity:object> tags are drawn from the Activity Streams specification, and are the portions of the entry that applications should program against.

```
<entry>
  <title>Snapshot Smith uploaded a photo.</title>
  <id>http://www.facebook.com/album.php?aid=6&id=499225643&ref=at</id>
  <link href="http://www.facebook.com/album.php?aid=6&id=499225643&ref=at" />

[MSV] I think there's a bug here (it's using &amp; instead of just &).  Look at wiki source if what I'm saying doesn't make sense ...

  <published>2009-04-06T21:23:00-07:00</published>
  <updated>2009-04-06T21:23:00-07:00</updated>
  <author>
    <name>Snapshot Smith</name>
    <uri>http://www.facebook.com/people/Snapshot-Smith/499225643</uri>
  </author>
  <category term="Upload Photos" label="Upload Photos" />
```

```
[MSV] What is this?  Is this the album name?  Why isn't this just Photos?

  <activity:verb>
    http://activitystrea.ms/schema/1.0/post/
  </activity:verb>
  <activity:object>
    <id>http://www.facebook.com/photo.php?pid=28&id=499225643&ref=at</id>
    <thumbnail>http://photos-e.ak.fbcdn.net/photos-ak-snc1/v2692/195/117/499225643/s499225643_28_6861716.jpg</thumbnail>
    <caption>A very attractive wall, indeed.</caption>
    <published>2009-04-06T21:23:00-07:00</published>
    <link rel="alternate" type="text/html" href="http://www.facebook.com/photo.php?pid=28&id=499225643&ref=at" />
    <activity:object-type>
      http://activitystrea.ms/schema/1.0/photo/
    </activity:object-type>
  </activity:object>
</entry>
```

The next entry represents a new note (with photo attachment) that a user, Nota Bene, wrote and posted to her profile. The entry includes two self-identifying <activity:object> entries, the first describing the note itself, the second describing the photo attachment. As with the first example, the assumption is that existing Atom readers can digest and render content based on the standard Atom tags, and Activity Streams-aware applications can program against the <activity:verb> and <activity:object> entries.

```
<entry>
  <title>Nota Bene wrote a new note: Here&apos;s a new note.</title>
  <id>http://www.facebook.com/note.php?note_id=71154267806&ref=at</id>
  <link href="http://www.facebook.com/note.php?note_id=71154267806&ref=at" />
  <link rel="via" href="http://www.facebook.com/note.php?note_id=71154267806&ref=at" />
  <link rel="related" href="http://www.facebook.com/note.php?note_id=71154267806&ref=at" />
  <published>2009-04-08T19:29:53-04:00</published>
  <updated>2009-04-08T19:29:53-04:00</updated>
  <author>
    <name>Nota Bene</name>
    <uri>http://www.jcain.devrs005.facebook.com/people/Nota-Bene/499225638</uri>
  </author>
  <category term="Create Note" label="Create Note" />

[MSV] Same question about 'Create Note'.

  <activity:verb>
    http://activitystrea.ms/schema/1.0/post/
  </activity:verb>
  <activity:object>
    <id>http://www.facebook.com/note.php?note_id=71154267806&ref=at</id>
    <title>Here&apos;s a new note.</title>
    <content>
      Here&apos;s a new note, everyone.  It&apos;s viewable by everyone.
    </content>
    <published>2009-04-08T19:29:53-04:00</published>
    <link rel="alternate" type="text/html" href="http://www.facebook.com/note.php?note_id=71154267806&ref=at" />
    <activity:object-type>
      http://activitystrea.ms/schema/1.0/note/
    </activity:object-type>
  </activity:object>
  <activity:object>
    <id>http://photos-g.ak.fbcdn.net/photos-ak-snc1/v2687/232/18/499225638/s499225638_30_6831075.jpg</id>
    <thumbnail>http://photos-g.ak.fbcdn.net/photos-ak-snc1/v2687/232/18/499225638/s499225638_30_6831075.jpg</thumbnail>
    <published>2009-04-08T19:29:53-04:00</published>
    <link rel="alternate" type="text/html" href="http://photos-g.ak.fbcdn.net/photos-ak-snc1/v2687/232/18/499225638/s499225638_30_6831075.jpg" />
    <activity:object-type>
      http://activitystrea.ms/schema/1.0/photo/
    </activity:object-type>
  </activity:object>
  <content type="html">Here&apos;s a new note, everyone.  It&apos;s viewable by everyone.</content>
</entry>
```

Seeing a Sample Syndicated Stream

You can take a look at a sample Activity Stream feed from a Facebook test user account. Pass user ID 499225637 as the source_id URL parameter. You don't need to pass the app_id and the sig parameters. That is, go to http://www.facebook.com/activitystreams/feed.php?source_id=499225637 in your browser. You can't write any content into this test user's stream, but you can get any content from here.