Report on Tracking/Update Options for Flash

Jonathan Olson

October 30, 2008

Introduction

Ticket Description

We have discussed many possibilities regarding tracking & updates for Flash simulations, but don't know about some of the technical issues involved. Here are the features we use on the Java side that we would like to investigate for the Flash sims:

1. Connecting to and reading from a URL to identify the latest version of a simulation
2. Opening a web browser to the specified URL if an update is available
3. Saving preference information locally (regarding whether the user allows tracking & updates)
4. Sending tracking information to a web service.

We'd like to know what is possible both for running Flash sims online (i.e. from our website) and for offline (i.e. when the user downloads a JAR file that has a SWF bundled inside it). Please prepare a report about the 8 issues above (1-4 for both online and offline use), and feel free to create prototypes to help identify and explore issues.

Background

Local Security Considerations

In Flash Player 8 and later, new security restrictions were announced that restrict applications being run locally (either in a browser or stand-alone player) that affect the Offline and CD distribution of simulations.
(About local file security and Flash Player – Version 8).
Without explicitly setting the application as trusted (Figure 1), it can either **access the local file system OR the network, but not both**, which prevents malicious Flash applications running locally from reading sensitive files and relaying them to online agents. This decision is made when the Flash application is published within the Publish Settings dialog (Figure 2). Since the current Flash simulations read a local xml file for internationalization information, for tracking either the internationalization (and other external data if it exists) will need to be passed to the .swf without reading local files, or the Flash part of the simulation will not be able to send/receive any information from the network. This includes pointing browsers to the phet.colorado.edu page,

Figure 1: Flash Player Global Security Settings: To allow a local application access to both files and the network, its location must be added through here.

reading URLs, and sending any sort of information. Discussion has indicated that passing the internationalization information into the simulation would be the best option, and will be discussed later on.

Networking Security Considerations

If a simulation is run online from phet.colorado.edu, it will automatically be able to send and receive data from phet.colorado.edu, and point a browser to open a window/tab to an updates page. However when run locally (CD or Offline) in the network security sandbox, phet.colorado.edu is considered an untrusted domain. The Flash application should first allow communication with phet.colorado.edu:

```
System.security.allowDomain("phet.colorado.edu");
```

(About domains, cross-domain security, and SWF files – Version 8).
The Flash Player will then request phet.colorado.edu/crossdomain.xml (or a different file specified by System.security.loadPolicyFile()), to check whether the current domain (localhost) is allowed. This sample crossdomain.xml file will allow access:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy
  SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

Proof of Concepts for Communication and Preferences

Reading Current Version From URL

(Test XML Read, Source Code (fla)).
ActionScript 2 is able to load XML from remote sources, given that the security concerns above are satisfied (Simulation run online from server, or offline in network security sandbox). The following code is an example of reading a version number from a remotely-requested XML file:

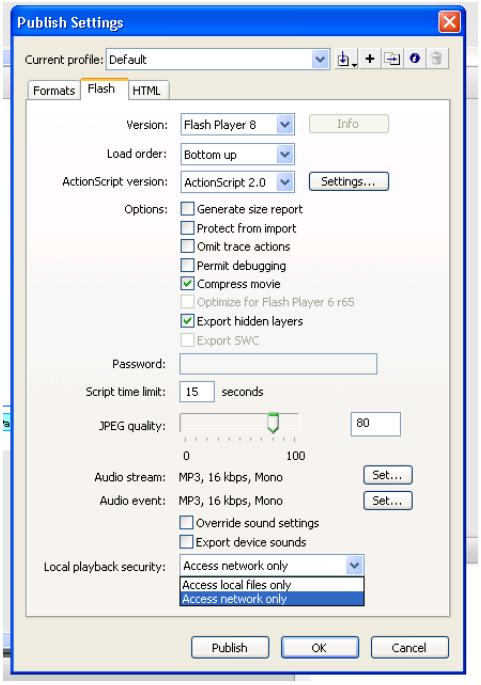

Figure 2: Dialog to change the security sandbox settings.

```
System.security.allowDomain("phet.colorado.edu");

var test_xml : XML = new XML();

// IMPORTANT: otherwise newlines and other
// whitespace are considered XML nodes.
test_xml.ignoreWhite = true;

// callback function called when XML load either
// succeeds or fails
test_xml.onLoad = function(success : Boolean) {
    if(success) {
        output_text.text += "\nSuccessfully downloaded and parsed:\n";
        output_text.text += test_xml.toString();
        output_text.text += "\n\nDetected version: ";
        output_text.text += test_xml.firstChild.firstChild.firstChild.attributes.version;
    } else {
        output_text.text += "\nFailure\n";
    }
}
output_text.text = "Attempting to read from ";
output_text.text += "http://phet.colorado.edu/tracking/flash-tests/test.xml:\n";
test_xml.load("http://phet.colorado.edu/tracking/flash-tests/test.xml");
```

Simulations would be able to request the current version by requesting an XML document particular to that simulation. This code will function properly with all methods that people can use to view simulations (online and offline), and if the internet is not accessible it will be caught and print "Failure". On success, it will print:

```
Attempting to read from http://phet.colorado.edu/tracking/flash-tests/test.xml:
```

```
Successfully downloaded and parsed:
<xml version="1.0"><test><testnode version="1.06.05" /></test></xml>

Detected version: 1.06.05
```

Opening Location with Web Browser

(Test Page Open, Source Code (fla)).
Flash has a few options (opening in new window/tab or replacing current window/tab), but again, the security considerations need to be taken into account to open a location in a web browser for updates (it is considered communicating with the network). Hopefully the code should be self-explanatory:

```
System.security.allowDomain("phet.colorado.edu");

// open a new window or tab for the target
getURL("http://phet.colorado.edu", "_blank");

// replace the current window or tab with the target
getURL("http://phet.colorado.edu", "_self");
```

It can open arbitrary URLs under phet.colorado.edu, for however the update system will work. Again, this will work for online and offline usage, as long as no local files are read.

Saving Preferences Locally

(Test Preferences, Source Code (fla)).
This feature will work regardless of the security sandbox. By default, Flash Player allows each domain to store 100 kB of data in "Local Shared Objects" for saving preferences and tracking sessions. Any data in Flash can be stored in local shared objects, and the prototype demonstrates storage of whether tracking is allowed, and the number of times the application has been started. This type of data can be checked when any simulation is started, and any simulations running under the same domain (either localhost or phet.colorado.edu) can share the same data. This may present a problem: If a user selects to disallow tracking on an online simulation (domain phet.colorado.edu), tracking will not be disabled if the user runs a simulation offline (domain localhost).

Sending Tracking Information

(Test Send XML, Source Code (fla)).
Just as ActionScript 2 can easily load remote XML documents, it can also send XML to a server (the preferred way of server communication in Flash). As for loading XML, security must allow the simulation to access the network. The XML.send() method sends XML in either GET or POST, however it opens up server's response in either the same or a new window/tab. The preferred method for sending tracking information would be to use XML.sendAndLoad(), which sends XML to the server (GET or POST), and parses the server's XML response. The prototype sends the following XML code to the server:

```
<tracking sessionId="1095834" type="session-started">Other Data</tracking>
```

Which is handled by PHP:

```
<wrapper><?php
        $xml = simplexml_load_string($HTTP_RAW_POST_DATA);
        echo 'Received from session ' . $xml['sessionId'];
?>
</wrapper>
```

And returns the result back to Flash. The application will print out:

```
<tracking sessionId="1095834" type="session-started">Other Data</tracking>
Sending to http://phet.colorado.edu/tracking/flash-tests/xml_post.php

Success, should be session id enclosed in wrapper tags!

<wrapper>Received from session 1095834</wrapper>
```

The XML can be traversed in PHP fairly easily using PHP 5's simplexml in this case, and PHP can talk directly to MySQL to store the tracking information.

Flash Tracking Information

(Test Info, Source Code (fla)).
There are certain types of tracking information that are somewhat Flash-specific, and some information that cannot be detected from within the Flash environment. Flash and Java both have data that is specific, like version numbers. However a number of fields for Java tracking are either detectable in Flash in a slightly different form, or not detectable in Flash. For instance, Flash can identify the language code (en, fr, ...), but is not given the country code, whereas Java is given en-US. Flash cannot detect the architecture, and gives different strings for operating system information, which will have to be either reconciled to fit the same format before insertion into the database, or have queries designed to handle both types of data when analyzing statistics. The link above is a sample of what information is available directly to the Flash player, without considering Javascript communication. Here is an example for two systems:

```
System.capabilities.hasAudio: true
System.capabilities.hasAccessibility: false
System.capabilities.language: en
System.capabilities.manufacturer: Macromedia Linux
System.capabilities.os: Linux 2.6.20-17-generic
System.capabilities.playerType: PlugIn
System.capabilities.screenResolutionX: 1920
System.capabilities.screenResolutionY: 1200
System.capabilities.version: LNX 9,0,124,0
Time zone offset / 60 (hours): 6

System.capabilities.hasAudio: true
System.capabilities.hasAccessibility: false
System.capabilities.language: en
System.capabilities.manufacturer: Macromedia Windows
System.capabilities.os: Windows XP
System.capabilities.playerType: PlugIn
System.capabilities.screenResolutionX: 1536
System.capabilities.screenResolutionY: 960
System.capabilities.version: WIN 9,0,45,0
Time zone offset / 60 (hours): 6
```

This can be compared to a sample of corresponding fields from the current Java tracker:

```
sim-locale=en_US
os_name=Mac OS X
os_version=10.5.5
os_arch=i386
javawebstart_version=null
java_version=1.5.0_16
java_vendor=Apple Inc.
user_country=US
user_timezone=America/Denver
locale-default=en_US
```

Conclusion

Tracking from Flash should be feasible in general, for both the online and offline versions. External data files will need to be integrated into the .swf or passed to the application in other ways for offline tracking to work, but is doable by a number of possible methods. XML seems to be the easiest way to transfer data between Flash and a server running PHP ≥ 5, and depending on the difficulty of XML on the Java side, might be best for all tracking communication.

Future Work

Internationalization and Other External Files

Getting internationalization data into the simulations without using external files is essential for tracking. FlashVars is a possibility, which feeds attributes in PARAM and EMBED tags in the html to local variables in the Flash simulation. Encoding (and escaping?) the XML to fit in an attribute needs to be investigated for FlashVars, and script generation of the .html files may be necessary (internationalization XML would seemingly need to be duplicated for both the PARAM and EMBED tags. Maybe there is a Javascript workaround?). Using ActionScript-Javascript communication with ExternalInterface would be able to pass in a string with less encoding issues possibly, but declaring the string in Javascript will probably be subject to the same escaping. Size should not be an issue, and the string won't need to be duplicated, however if a script is needed to generate HTML anyways, is it worth it? A slight modification on FlashVars should also be possible, as .swf files can handle query strings (sim.swf?internationalization=...), but may be very similar in use.

Tracking Information and Format

Query string-format variable passing may be more difficult than XML for Java? XML seems to be a good solution for Flash communication, but the format of passed information should be discussed. Additionally, the differences between Flash and Java tracking information, and how it should be sent is interrelated with how it is processed with PHP and stored in the database.