

## FAQ

- [Q1 - Where do I track browser implementation support of the HTML5 FileAPI?](#)
- [Q2 - Site works with HTTP but not with HTTPS, unclear Flash errors.](#)
- [Q3 - FireFox crashes with large downloads?](#)
- [Q4 - In the logfile I see: \*\*"Error: \[8\] Undefined index: </AttributeName/>"\*\*. What does it mean?](#)
- [Q5 - Does FileSender have more than one authentication mode?](#)
- [Q6 - How to quickly create big test files?](#)
- [Q7 - Upload stops after 8 hours?](#)
- [Q8 - /etc/cron.daily/filesender PHP Deprecated warning on Ubuntu?](#)
- [Q9 - Uploads are relatively slow?](#)

---

### Q1 - Where do I track browser implementation support of the HTML5 FileAPI?

**A - Check the ["When can I use" website](#) to monitor implementation progress of the HTML5 FileAPI for all major browsers.** In particular support for [FileReader API](#) and [Blob URLs](#) needs to be light green (=supported) for a browser to support uploads larger than 2GB

---

### Q2 - Site works with HTTP but not with HTTPS, unclear Flash errors.

Observed behaviour:

Flash error pop-ups like ["#2038 IO Error"](#) but also other error numbers have been observed.  
Also:

I can log in, fill in blank fields, select file to upload, but when I click send, I immediately get the message:

```
Uploading ubuntu-9.10-server-amd64.iso...
File uploaded
Moving File
```

The progress bar remains 0%.

**A - This problem usually occurs when a self-signed SSL server certificate is used which Flash doesn't like.**

When Flash wants to verify a certificate it looks in the OS specific system certificate store for a trusted 'root' certificate to verify against (note that when using for example FireFox or Opera this will **not** be the certificate store used by the browser). When using a self signed certificate there most likely won't be a trusted root certificate. You can ask your users to put your self signed certificate in the 'system certificate store' (for Windows the MS "Trusted Root Certification Authorities" store and for Mac OSX in the "system key chain"). After that Flash, even when used with Firefox or Opera seems to be happy when uploading over SSL. A better solution is to get a real SSL server certificate that's recognised by all major operating systems.

---

### Q3 - FireFox crashes with large downloads?

**A - Using FireFox**

Disable the **FireBug** plugin as it causes FireFox to store the download in memory and will crash when you run out of RAM.

---

### Q4 - In the logfile I see: **"Error: [8] Undefined index: </AttributeName/>"**. What does it mean?

**A - You likely need to tell SimpleSamlPHP how to handle certain attributes**

First, check if the IdP is sending the attribute mentioned by `/AttributeName/`.  
If no, ask the IdP administrator to add this attribute to the allowed-to-send list.  
If yes, check which format the IdP uses to send the attributes. It will be either `oid-` or `urn-format`.  
By default, simplesamlphp is configured to translate from "oid to attribute name".  
As many IdPs use urn-format only, it is recommended to enable the translation from "urn to attribute name" as well.  
You can do this by adding the text below in bold to the simplesamlphp configuration file (i.e. `../simplesamlphp/config/config.php`)

```
/*
 * Authentication processing filters that will be executed for all IdPs
 * Both Shibboleth and SAML 2.0
 */
'authproc.sp' => array(
    10 => array(
        'class' => 'core:AttributeMap', 'oid2name'
    ),
    *15 => array(
        'class' => 'core:AttributeMap', 'urn2name'
    ), *
```

---

### Q5 - Does FileSender have more than one authentication mode?

**A - Users can authenticate in two ways to the system:**

1. through one of the supported AuthN mechanisms
2. using an upload voucher that an authenticated user gave them

---

#### Q6 - How to quickly create big test files?

**A - On a Windows machine this is straight forward:** to create a file of a certain size and if the file contents don't matter, you can use the Fsutil command as follows:

```
fsutil file createnew <name of file> <size in bytes>
```

For example,

```
fsutil file createnew d:\temp\10GBfile.txt 10737418240
```

creates a 10GB file named 10GBfile.txt in the d:\temp folder.

More practical examples:

```
D:\>fsutil file createnew 90GBfile.txt 96636764160
File D:\90GBfile.txt is created
```

```
D:\>fsutil file createnew 4.7GBfile.txt 504658652
File D:\4.7GBfile.txt is created
```

```
D:\>fsutil file createnew 1GBfile.txt 1073741824
File D:\1GBfile.txt is created
```

```
D:\>fsutil file createnew 100MBfile.txt 104857000
File D:\100MBfile.txt is created
```

```
D:\>fsutil file createnew 500MBfile.txt 524288000
File D:\500MBfile.txt is created
```

```
D:\>fsutil file createnew 50MBfile.txt 52428800
File D:\50MBfile.txt is created
```

```
D:\>fsutil file createnew 5MBfile.txt 5242880
File D:\5MBfile.txt is created
```

#### A - On a Mac and Linux

In a terminal window

```
dd if=/dev/urandom of=testfile.dat bs=1024 count=1024
```

'if' is InputFile, 'of' is OutputFile, 'bs' is BlockSize (in bytes) and 'count' is the number of blocks (of size 'bs') to write. As input in this example you get random data (from the pseudofile /dev/urandom). You could speed up the process a little by creating a file with 'zeroes'. In that case you can use 'if=/dev/zero'. I prefer 'random' data because it is more 'realistic' data (a file of only zeroes can and will be highly compressed with up- and downloads, random data resembles more an already compressed file like a zip-archive).

---

#### Q7 - Upload stops after 8 hours?

**A: The standard session duration timeout of SimpleSamlPHP is 8 hours.** When uploading using Gears, the PHP code at the backend checks the authentication status for every uploaded chunk. If the session is longer then the session timeout of SimpleSamlPHP, the upload stops. Increase the value of the session duration variable in your simplesaml config.php (session.duration) to a value that will cover the estimated upload time period.

---

#### Q8 - /etc/cron.daily/filesender PHP Deprecated warning on Ubuntu?

On Ubuntu the daily cronjob reports:

```
/etc/cron.daily/filesender: PHP Deprecated: Comments starting with '#' are deprecated in /etc/php5/cli/conf.d/mcrypt.ini on line 1 in Unknown on line 0
```

**A - This is a bug in the PHP version on Ubuntu.** See

<https://bugs.launchpad.net/ubuntu/+source/php-mcrypt/+bug/540208>

Until the bug is fixed just replace # with ; in /etc/php5/conf.d/mcrypt.ini.

---

#### Q9 - Uploads are relatively slow?

Uploading large files is relatively slow when compared with downloading. Can this be improved?

**A - There are a few things that might help:**

- Check your debug settings in config.php. Disabling debug logging and client\_specific\_logging will in general improve upload performance:

```
$config["debug"] = false;
$config["client_specific_logging"] = false;
```

- For improving the Gears upload you can increase the CHUNK\_BYTES parameter in www/js/fs\_gears.js (requires manual edit)

```
var CHUNK_BYTES          = 2000000;          // 2MB (default is
2000000 = 2MB)
```

Values larger than 2 to 5 MB will not make much difference anymore and will make the

upload progress bar taking less and larger steps.