

Technical Details in OpenLSD

Author: Frédéric Brégier under LGPL

SIMILARITY CHECK IN OPENLSD	2
OPENLSD.SERVER.ADMIN.CHECKSIMILAR	2
<i>Crypted Legacy</i>	2
<i>From outside the OpenLSD Server</i>	2

Website: <http://openlsd.free.fr/en/OpenLSD.html>

Similarity Check in OpenLSD

openlsd.server.admin.CheckSimilar

For some specific applications, users want to know before the insertion of the document in the OpenLSD system if this document is already in.

From my point of view, the first step should be always the business approach that is to say to check if the same document (from the business point of view) is already in the system. To do that, comparing the business index of this new document and the database should be enough. Therefore in this case, one is using only the database of the business application.

This approach is the default one in OpenLSD. Two documents cannot have (in the same Legacy) the same business index.

However, when this is not enough, the application can use the new function: **CheckSimilar**.

This function works only on the OpenLSD Server since the tested document must be local to the Legacy where it should be imported, due to its implementation.

This function checks if there is at least one document with the same size and “empreint” (MD5 checksum), and for each of them, it compares the stored file with the source one with a byte to byte comparison. If there is at least one existing document, this function returns the corresponding LSD index and a status of 1. If none exists, it has a status of 0.

This function can be chained with an import in a script, using the returned status to choose to import or not this file.

Another way is to use the new import functions which test the existence of the future imported files, before importing the files. This function is proposed as **LSDServerImportXXX** functions with the extension “**CheckSimilar**”. “**Server**” means that it can be only run on the OpenLSD Server which hosts the Legacy. “**XXX**” are the different kind of options of import (**BLOCK** or not, **ML** or not).

Crypted Legacy

Those functions do not work on crypted Legacy since the MD5 and the size are not the same between the original file and the crypted stored file.

However, one can implement a similar function by doing this way:

- First crypt the source file.
- Second applies the function **CheckSimilar** (not import functions).

This is possible only if the crypted file is always the same whatever the date the file is crypted (that is to say the crypto algorithm must not depend on the current date).

From outside the OpenLSD Server

If one needs this function to be running outside the OpenLSD Server, one could consider this scenario:

- First upload the file onto the OpenLSD server,
- Launch the **CheckSimilar** and then have a response back to the client which can consider to import or not the document (two transfers are implied)
- Or use one of the **LSDServerImportXXX** with the extension “**CheckSimilar**” after the upload but without forgetting to delete the source file if not imported.