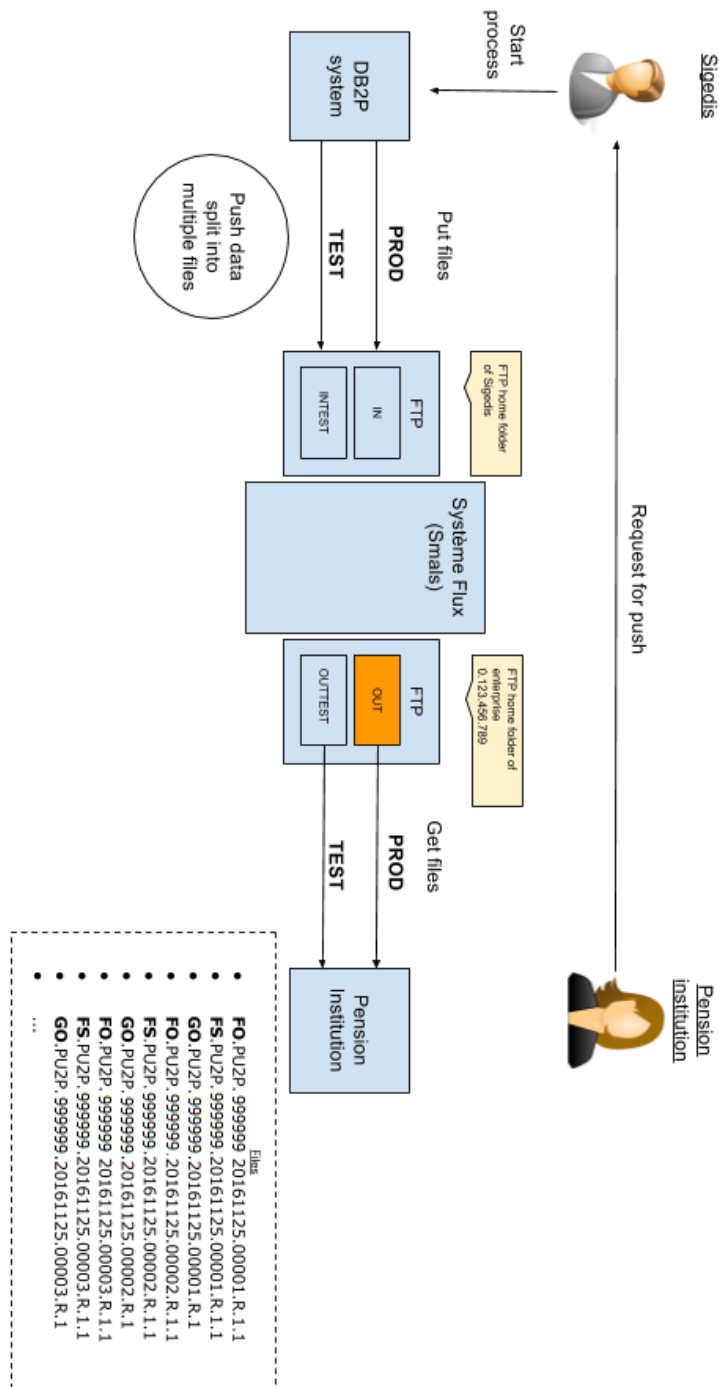


<b>Concerne</b>	Sigedis-DB2P- push flux - Technical specification
<b>Date</b>	20/06/2017
<b>Version</b>	V3

## INTRODUCTION



## CONCEPTS

### Data files

A push operation consists in extracting the data from the DB2P system into large XML (utf-8 encoded) batch files. These files are hereunder referred to as "data files".

When the result of an extraction is too large to hold into a single data file, multiple data files are created.

### Système Flux

Système Flux is the piece of middleware that enables Sigedis to exchange files with pension institutions. It is operated and supported by Smals.

## DATA TRANSFER

---

As stated above, the data extracted from the DB2P system are stored into one or multiple large (aka "batch files") XML data files. These files then need to be sent to the pension institution.

This is done using Smals' Système Flux system by using FTPs.

The data files are sent according to Système Flux's protocol. Following this protocol, each data file will be accompanied by two other "technical" files.

So, for each data file, there will be 3 files :

1. The data file itself with a name that will look like **FO**.PU2P.999999.20161125.00001.R.1.1
2. Next to it, a "signature file", named, for example, **FS**.PU2P.999999.20161125.00001.R.1.1. This file includes a signature of the data file. It can prove that the file has not been tampered with. It is recommended that the pension institution validates it.
3. Finally, there will also be a GO file named, for example, **GO**.PU2P.999999.20161125.00001.R.1, which is a marker file and whose presence indicates that a data file has been completely put on the FTP and is ready for retrieval. Consequently, from this moment, the pension institution can start retrieving the corresponding data file. This file's sole purpose is to avoid having the pension institution start to retrieve a file that is not yet completely placed on the FTP by Système Flux.

Notes :

- **none of these files have a file extension.**
- All files are intended for the pension institution's use. Therefore, to keep things clean and safe, Sigedis recommends that the pension institution removes them after consumption, even if file name conventions should avoid any conflicts with leftover files. However, Système Flux will automatically remove files older than 6 months.

## NAMING CONVENTION

The Système Flux files are named according to the following naming convention :

for a file named "FO.PU2P.999999.20161125.00001.R.1.1" :

- FO (or FS or GO (see previous section)) : the type of the file
- PU2P : content indication. Here a push flux for DB2P
- 999999 is the sender id
- 20161125 is the creation date of the file
- 00001 is a sequence number for files sent the same day (that is, with the same creation date)
- R is for Production data, T is for Test data
- 1.1 is the number of parts followed by the part number. This is not used in the context of the DB2P Push Flux, so all FO files will end with "1.1".

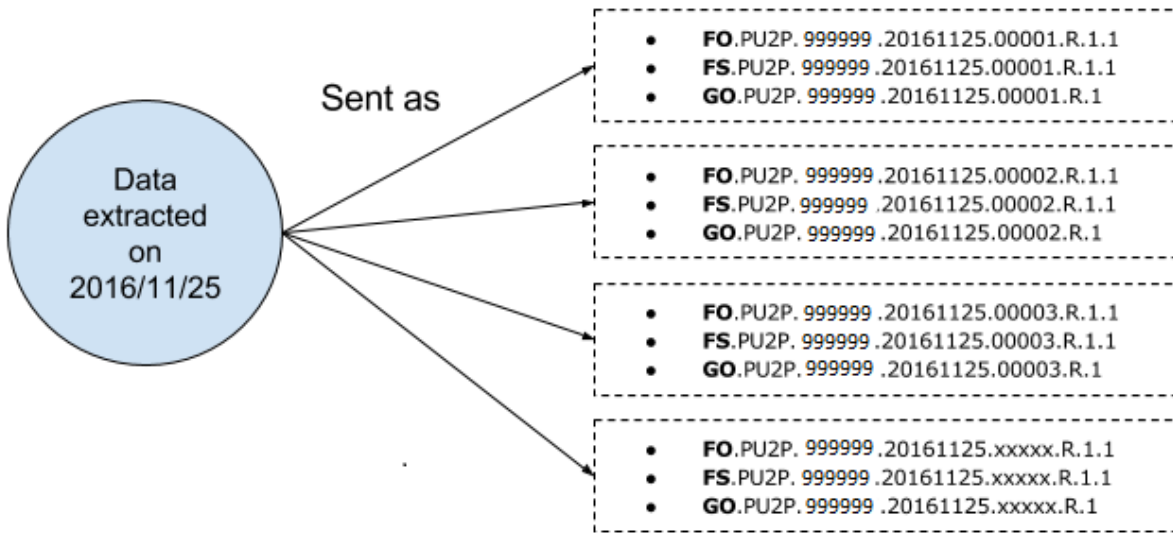
The name of the FS file follows the same rules. That of the GO file too, except that there is no part number in it.

## MULTIPLE DATA FILES

The amount of push data that must be sent to the pension institution might be so large that it cannot fit into a single data file. The maximum size of a data file cannot be given precisely. The only hint that is safe to write is that if it were zipped, its size would never exceed 95 MB. This limit comes from the fact that Sigedis transmits the data file to Système Flux in a compressed format and it is not allowed to transmit zip files with a size that exceeds 95 MB.

When the push data set must be sent as multiple data files, it will result in multiple FO files whose filename will only differ on the "sequence number" part.

Visually, such a situation can be represented as follows :



Note : by only looking at the filenames there is no way to determine that two FO files are part of the same push flux data set. If multiple push flux data sets (pension fiches and then affiliate data for example) need to be sent the same day, it is not possible to tell which files are part of one or another by solely inspecting the filenames. To do that, the content of the file needs to be inspected. It will tell which what type of data it holds and which part of it.