# DAILY CLIENT PDF OUTPUT EXERCISE (also Quarterly as of January 2018)

Last Updated: 1/19/2018 by Brian Williams

Note: This job is not required for all clients. SCAN is currently in Production with it; Meridian and Healthspring are the next 2 required clients. There are 2 processes in one in this folder

* Process 1 – reads the PDF data from the database fields and produces the output (CURRENT as of 11/9/2017). This will NOT be used as we want to get rid of the pdf column in PostGres (DEAD)
* Process 2 – reads the PDF data from files on the SFTP server and produces the output (LIVE in Q1 2018). Package names end in “NEW\_FROM\_FTP”

# Required Elements:

Access to [\\sqlrdp\general$\temp](file:///\\sqlrdp\general$\temp) directory - underneath these the ability to create/manage folders per client (e.g., [\\sqlrdp\general$\temp\scan](file:///\\sqlrdp\general$\temp\scan) or [\\sqlrdp\general$\temp\healthspring](file:///\\sqlrdp\general$\temp\healthspring), etc.). These files will be removed upon job completion.

Access to the 3 MedPro PostgreSQL environments to run queries to gather communications data from database (looks at previous day to today) – *but we need the ability to tweak this on command to be 2 days, 3 days, etc. if failures occur*

Access to the configuration files – if you need to test this (these are the .dtsConfig files)

# Steps

Step 1: Run a query that selects communications data elements for a particular client within the specified timeframe (typically previous day). Something like this should work against PostGres

select Distinct pt.member\_id , c.label as description , cast(pdf.communication\_id as character varying (100)) || '.pdf' as fileName,pdf.created\_at as createdAt,cast (current\_date as timestamp without time zone) as publishDate from communication\_pdfs pdf inner join communications c on pdf.communication\_id = c.id inner join patients pt on c.patient\_id = pt.id inner join organizations o on pt.organization\_id=o.id where (pdf.created\_at between cast(CURRENT\_DATE-1 as date) and cast(CURRENT\_DATE as date) or c.id in (select id from x\_communication\_export)) and o.pmd\_client\_id = 62;

Step 2. A file is created listing the manifest of files that are to be included in the PDF extract. This file is dropped in the temp directory listed above. ([\\sqlrdp\general$\temp\scan\pdf](file:///\\sqlrdp\general$\temp\scan\pdf) or [\\sqlrdp\general$\temp\highmark\pdf](file:///\\sqlrdp\general$\temp\highmark\pdf), etc.)

Step 3: The OLD process generates the PDF from data within the database and writes it to the temp directory. Again, the temp directory is temporary and is cleared upon job completion. ([\\sqlrdp\general$\temp\scan\pdf](file:///\\sqlrdp\general$\temp\scan\pdf) or [\\sqlrdp\general$\temp\highmark\pdf](file:///\\sqlrdp\general$\temp\highmark\pdf), etc.). NOTE: The NEW version will pull from documents stored on the SFTP server in a date/time folder breakdown structure.

Step 4. The PDF files are compressed and zipped into a file containing all data in the pdf directory. The zip file is then pushed to the PharmMD FTP server using the datadumpftp account for the appropriate default remote directory (scanhealth, Healthspring, etc.)

Step 5. The temp PDF directory listed above is cleared out so the run is clean for the next day.

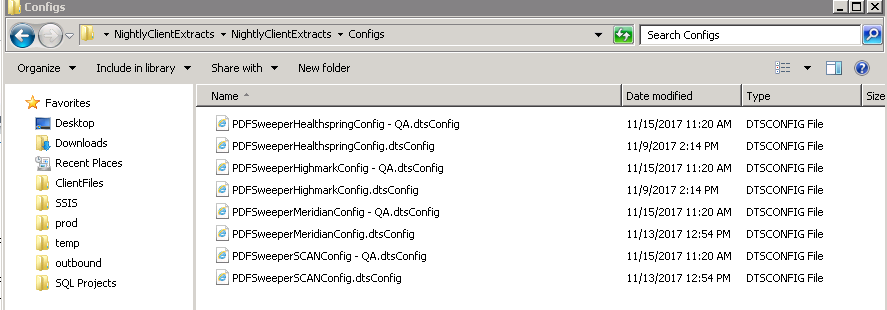
Step 6. Any leftover files in the client specific temp directory are moved to the processed folder within the Client Files folder on SQLRDP ([\\sqlrdp\general$\ClientFiles\Scan\Processed\\_daily\outbound](file:///\\sqlrdp\general$\ClientFiles\Scan\Processed\_daily\outbound) or [\\sqlrdp\general$\ClientFiles\Highmark \Processed\\_daily\outbound](file:///\\sqlrdp\general$\ClientFiles\Highmark%20\Processed\_daily\outbound), etc.) For NON Production runs we use the following directory structure (ClientFiles\\_ClientFiles\_Y). Examples are [\\sqlrdp\general$\ClientFiles\\_ClientFiles\_y\Scan\Processed\\_daily\outbound](file:///\\sqlrdp\general$\ClientFiles\_ClientFiles_y\Scan\Processed\_daily\outbound) or [\\sqlrdp\general$\ClientFiles\\_ClientFiles\_y\Highmark \Processed\\_daily\outbound](file:///\\sqlrdp\general$\ClientFiles\_ClientFiles_y\Highmark%20\Processed\_daily\outbound))

# NOTES:

Each client has a specific configuration file that should be used with them. These are in a folder under here called “Configs”. Important variables are

* ClientCode - this should be a generic value that will be passed to write output file names and identify the client
* PGOrganizationName – the organization name within the tables in PostgreSQL (may or may not match the client code)
* SFTPDefaultDirectory – used by Jason Carroll and the network team for where the datadumpftp user should drop the files for client pickup or transmission
* DaysLookBack – this is typically 1 so the files sent would have data for the previous day only – you can override that behavior here
* OverrideYesterdayDefaultDate – by default getdate() is used within the application for the end date. You can modify that here if you wanted to run something for the previous quarter, year, etc.
  + By using an OverrideYesterdayDefaultDate of 9/30/2017 and entering a DaysLookback as 90, you would get data from 7/1/2017 to 9/30/2017 on your output files
* DisableSFTPOverride - if you want to turn OFF the SFTP’ing of the file to the datadumpftp folder turn this to TRUE. Otherwise, set it to FALSE
* IsProduction …. Pretty self explanatory. The only thing this controls of the SFTP directory/server that the files are dropped to. We also change the ClientFiles folder to \_ClientFiles\_y if we are not in production.

Additionally, there are configuration files for each client. One is Production and one is QA. The QA one does NOT do the ftp’d of files to the client (DisableSFTPOverride is set to true)

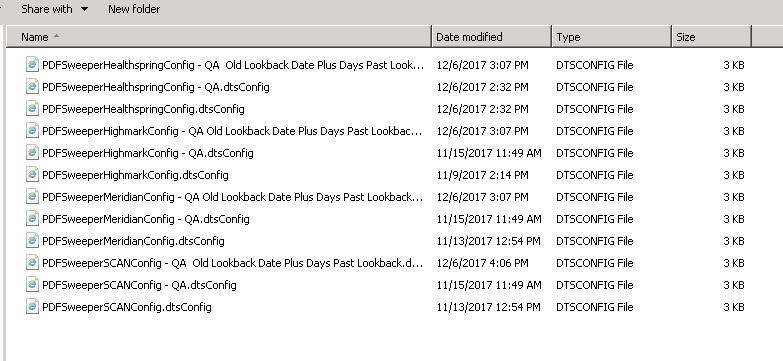
’

# TESTING NOTES

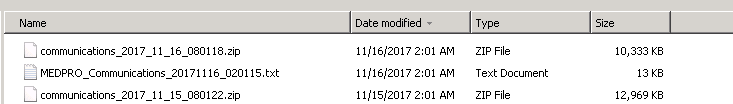
The packages are located here currently: [\\sqlrdp\G\artifacts\prod\com.pharmmd.outboundDataFeed.ClientGenericNightlyExtracts\NightlyClientExtracts\NightlyClientExtracts\bin](file:///\\sqlrdp\G\artifacts\prod\com.pharmmd.outboundDataFeed.ClientGenericNightlyExtracts\NightlyClientExtracts\NightlyClientExtracts\bin)

The QA config file of your choosing should be used to run the package Client\_Daily\_Outbound\_PDFS\_NEW\_FROM\_FTP.dtsx. This currently is only available for Meridian, Healthspring, Highmark, SCAN and Summacare (pending – January 2018).

The config files can be found here and look like the below([\\sqlrdp\artifacts\prod\com.pharmmd.outboundDataFeed.ClientGenericNightlyExtracts\NightlyClientExtracts\NightlyClientExtracts\Configs](file:///\\sqlrdp\artifacts\prod\com.pharmmd.outboundDataFeed.ClientGenericNightlyExtracts\NightlyClientExtracts\NightlyClientExtracts\Configs))



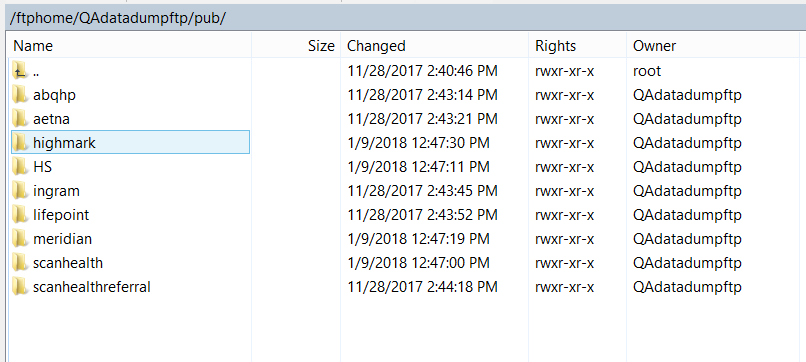
A communications\*.zip file and a MEDPRO\_Communications\*.txt file should be created after each specific run of the application.

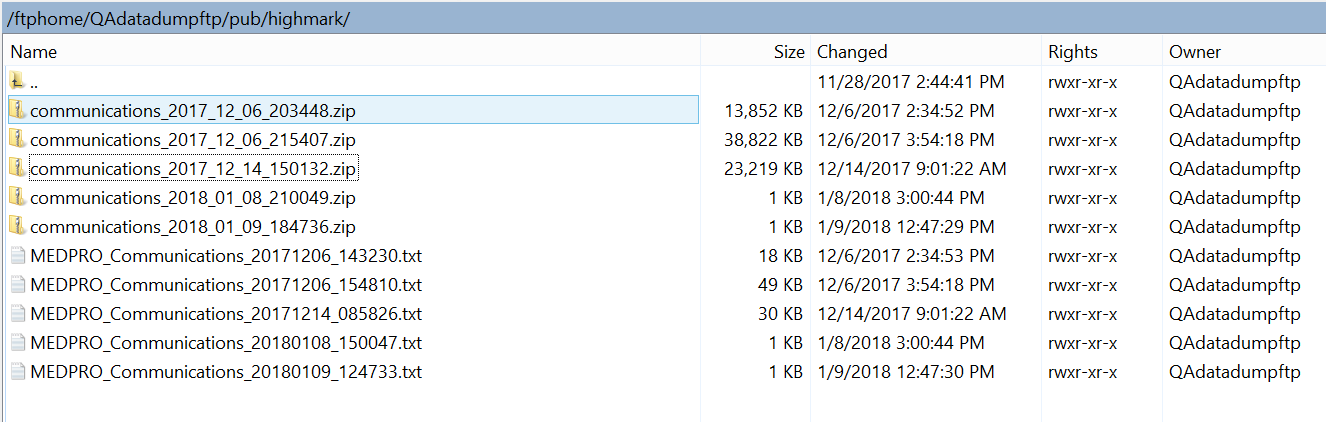


The files will be dropped on the QA SFTP that Jason Carroll set up for us.

* Meridian -> meridian
* SCAN – scanhealth
* Healthspring – HS
* Highmark – highmark
* Summacare - summacare

Here are the SFTP locations  
*QA*

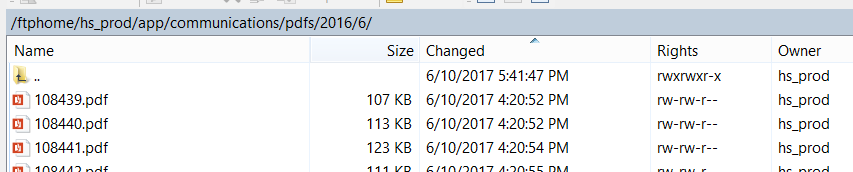




*PROD*

(can’t get screenshot due to lack of access) but it should follow the same client folder naming structure.

There will be a manifest test document that should coincide with the documents inside of the zipped up file. These will be (by default), communication documents from the previous day. These documents also reside on the SFTP server and are stored by year/month in the appropriate folder per client (example below for Healthspring for June 2016)



IMPORTANT NOTE: A variable in the QA config file ensures that these files are NOT sent out the client via the normal auto-Ftp’ing process. ([User::IsProduction]==False)

## Current Query for pulling PDFS – varies

select Distinct pt.member\_id , c.label as description , cast(pdf.communication\_id as character varying (100)) || '.pdf' as fileName , pdf.created\_at as sentDate ,cast (current\_date as timestamp) as publishDate , pdf.pdf from communication\_pdfs pdf inner join communications c on pdf.communication\_id = c.id inner join patients pt on c.patient\_id = pt.id inner join organizations o on pt.organization\_id=o.id where (pdf.created\_at between cast(**INPUTPREVIOUSDAY** as date)-**INPUTDAYSLOOKBACK** and cast(**INPUTPREVIOUSDAY** as date) or c.id in (select id from x\_communication\_export)) and UPPER(o.Name) like '%**INPUTCLIENTCODE**%'

# SPECIAL SECTION REGARDING QUARTERLY FILES

There is a slight weak to this process where we created a package called “Client\_Quarterly\_Outbound\_PDFS\_NEW\_FROM\_FTP.dtsx” to be used for quarterly PDF extracts of CMR Letters – this is specific to Meridian but can be applied to all clients. It just limits down the letter types and statuses of the communications we are sending. The files also go to a “\_quarterly” folder instead of the typical “\_daily” one on the file share.

This process will (by default) try to use the previous quarter fro m the current date but this can be overridden by a variable in the config files (OverridePreviousQuarter is the variable).

The syntax is as follows “Q42017” relates to the 4th quarter of 2017. “Q22016” relates to the 2nd quarter of 2016.

## Current Query for pulling only approved CMR Letters - varies

select Distinct pt.member\_id , c.label as description , cast(pdf.communication\_id as character varying (100)) || '.pdf' as fileName , pdf.created\_at as sentDate ,cast (current\_date as timestamp) as publishDate , pdf.pdf from communication\_pdfs pdf inner join communications c on pdf.communication\_id = c.id inner join patients pt on c.patient\_id = pt.id inner join organizations o on pt.organization\_id=o.id where c.delivery\_method='Letter' and c.status='approved' and c.endpoint\_type='Patient' and (pdf.created\_at between cast(**QUARTERTOUSESTARTDATE** as date) and cast(**QUARTERTOUSEENDDATE** as date) or c.id in (select id from x\_communication\_export)) " + @[User::PGClientLookupQuery]