request will have a new file extension addition (FILE_PROCESSING). Reads the OTFR request and creates Observation Status Files (OSFs) for each DATASET entry in the request (current plan is only one DATASET entry per request). The OSFs will be used to control and track the processing of the request while it is in the OTFR pipeline. A trailer (log) file is started that will document the processing steps performed by OTFR on this request.

The OTFR Request File will be named with a unique pseudo-TIMESTAMP and a DATASET_NAME, e.g. **3112234912345_u2440101t.req** . Since the DATASET_NAME could appear in more than one request, the TIMESTAMP must be guaranteed by DADS (who will be naming these files) to be unique for every OTFR request sent to the pipeline. The TIMESTAMP consists of several fields: `DDHHMMSSnnnnn`, where `DD` is the day of the month, `HHMMSS` are the hour, minute, and second at which the request was generated, and `nnnnn` is a unique number assigned by DADS that prevents duplicate timestamps. The OTFR request file will contain, at least,

```
DATASET_NAME=U2440101T
FILE_COUNT=0
TIMESTAMP=3112234912345
DIRECTORY=DISK$LEO:[LEO.MSWAM.OTFCAL.DADS]
END_FILE
```

When an OTFR request file is found in the polling directory (FILE_DIRECTORY1), a subdirectory named with the values of the TIMESTAMP_DATASET will be created under the calibration directory (OUTPATH) for this instrument. By grouping the subdirectories for each instrument under an instrument-specific directory, this allows segregation and disk management of processing by instrument.

If the subdirectory already exists, then this will be considered a duplicate request, and an OSF (Observation Status File) will be created for the request indicating the duplication ('d') in the PO column. The duplication OSF will also be given a special class (`CLS = req`) and a special DCF number (`DCF = 999`), so that these OSFs can be sorted together in the OMG. These entries on the OMG will hopefully catch the operator's attention. A message will also appear in the POLxxx process log file explaining the error. These duplications are likely only to be seen in the test environment, where the same OTFR requests can be run repeatedly using some tricks. DADS should not normally feed duplicate requests to the OTFR system.

The OTFR request file will be copied from the polling directory into this request-unique subdirectory. It is given a file extension addition (FILE_PROCESSING) at this time as an artifact of the OPUS architecture. This copy of the request will provide later pipeline stages access to needed information. For example, the DIRECTORY field in the request will need to be accessed in order to send the OTFR products back to DADS. The copy of the request in the local polling directory will be deleted automatically by the OPUS architecture to avoid processing the request multiple times.

The contents of the OTFR request are echoed to the process log file and the request trailer file, so that later searching of these log files can be used in request debugging. The request trailer file will be named **`TIMESTAMP_DATASET.trl`**, and will be created in the unique request-specific subdirectory. During processing, the OMG/ViewTrailer function should be able to display the contents of this file to indicate the steps performed so far in processing this request.

POLxxx will use the information in the OTFR request filename and the file contents to create OSFs on the OTFR blackboard. Here is an example OSF:

```
33C243B1-cw_____________________.31512014412345_u2440101t-wf2-000-____
```

The most relevant details here are the dataset identifier made up of the timestamp and datasetname (`31512014412345_u2440101t`) and the OSF status field (`cw_____________________`). These indicate to the OPUS architecture that a particular dataset is waiting (w) for a particular kind of processing. Since the 'w' is in the second pipeline stage, the LS (LSTxxx) process will be the next process to peform work on this request. The OSF status field is filled from the OSF_START_STATUS resource entry. The data_id field of the OSF (in this case set to