**Series of events**

**STEP - I**

* On the IS side; prf files (Index txt, print txt & pdf) need to be copied to: /d/isd\_share/<CID>/ folder in production; For parallel it will be in /z/isd\_share/<CID> directory.
* Naming convention:
  + - **Index txt proof file**:
      * The index txt file describe the context of the processing job and the txt & pdf file names.
      * /print\_fileid/{cid}{jid}{YYYYMMDDHHMMSS}.prf.txt
      * e.g.: /print\_fileid/afcums120170808130558.prf.txt
    - **Proof txt file:**
      * The TXT record describe the contents of the actual document
      * /print\_fileid/{cid}{jid}{seg}{cycle}.prf.{YYYYMMDDHHMMSS}
      * e.g.: /print\_fileid/afcums1015.prf.20170808130558
    - **PDF file:**
      * The PDF document file
      * /afpfiles/{cid}{jid}{seg}{cycle}.prf.pdf.{YYYYMMDDHHMMSS}
      * e.g.: /afpfiles/afcums1015.prf.pdf.20170808130558
* For the **new jobs** the following fields/flags need to be added to control file on print side:
  + olp = y/n
  + olp\_prf\_file = y/n
  + olp\_auto\_rel = y/n
  + olp\_int\_dlist = <grep ‘olp\_maillist ‘ from cidjid\_mail.lis>
  + olp\_divert = y/n
* For the **new jobs** the following fields/flags need to be added to index txt prf/full file:
  + Job\_2d = <3digit code>
  + olp = y/n
  + olp\_divert = y/n
* Pre-requisite: Existing flag changes in control file:
  + Delete or set job\_prf =’n’ as olp & job\_prf both cannot be set to ‘y’
  + mail\_full\_serv must be set to ‘ASE3’ or ‘ASE4’
* ~~Some of the field values will be hard coded in the index prf txt file; (e.g. account number, addr1, addr2 -~~ **~~# Anil/Rohit to confirm the fields based on Mongo table structure/ Rony to modify the code~~** - Per David & team none of the fields from index prf txt will be used on UI side hence no need to hard code any of the values, leave as is.
* If DT\_JOB\_ID is NOT equal to ‘**888888**’ or ‘**999999**’ then
* Use the following steps to determine “Post” (insert) or “Put/Patch” (update) call to **InfoSync StateUpdate** API:
  + Get the current job state by using **InfoSync GetState** API
* Parse the response from **InfoSync GetState** API based on returned olp\_id value. If olp\_id returns null then StateUpdate operation=Post else Put/Patch.
* Invoke **InfoSync StateUpdate** API using curl command w/ appropriate -X [Post or Put/Patch] parameter to insert/update the corresponding record.
* invoke Index file processor using:
  + **Prf file:** http:// pd-dev2.infoimage.com:8880/ indexFileProcess/trigger?host=10.8.9.60&username=pdev1&pwd=infoimage&path=/z/isd\_share/afcu/print\_fileid&filename= afcums1YYYYMMDDHHMMSS.prf.txt
  + **Full file:** http:// pd-dev2.infoimage.com:8880/ indexFileProcess/trigger?host=10.8.9.60&username=pdev1&pwd=infoimage&path=/z/isd\_share/afcu/print\_fileid&filename= afcums1YYYYMMDDHHMMSS.txt

**Note - ISD will use fields 427-436 to extract SSN from txt records if the submitted file is a TAX job.**

* If call to Index file processor returns ‘zero’ then
  + Invoke ETA Board API.
* Else have the process sleep of ‘30s’ & try again (total 3 times w/ ‘30s’ sleep) if still received the non-zero return code then
  + - Invoke **InfoSync StateUpdate** Patch API using curl command w/ state parameter as ‘**warning**’ to update the corresponding record.
    - Send a self-service email to olp\_int\_dlist mailing list (Similar to email that is generated if afp2pdf conversion fails)

**STEP – II**

* If Index file processor fails to insert records in MongoDB
  + Invoke **InfoSync StateUpdate** API with state = ‘warning’ ; type [prf|full]
    - Update [type]\_state & MAIN\_STATE columns to ‘warning’ & send an email notification to internal team
* If MongoDB insert is successful
  + Invoke **InfoSync StateUpdate** API state = ‘success’; type [prf|full].
  + Check if ‘olp = ‘Y’ AND ‘olp\_auto\_rel’ flag=’N’:
    - Calculate the state for MAIN\_STATE column
    - Update [type]\_state column to ‘success’ & MAIN\_STATE column with calculated State value (internal\_pending)
    - Send an email notification to internal team
  + Check if ‘olp = ‘Y’ AND ‘olp\_auto\_rel’ flag=’Y’:
    - Calculate the state for MAIN\_STATE column
    - Update [type]\_state column to ‘success’ & MAIN\_STATE column with calculated State value (customer\_pending)
    - Send an email notification to customer/internal team

**IS Database Schema Update:**

* Add columns to an existing **PROCESS\_MAIN** table to reference **PROCESS\_MAIN\_OLP** table:
  + **TABLE: PROCESS\_MAI**N

|  |  |  |
| --- | --- | --- |
| COLUMNS | TYPE | VALUE |
| OLP\_ID <FK> | NUMBER | 123 |

* Create **PROCESS\_MAIN\_OLP** table as an extension to **PROCESS\_MAIN** table:
  + **TABLE: PROCESS\_MAIN\_OLP**

|  |  |  |
| --- | --- | --- |
| COLUMNS | TYPE | VALUE |
| OLP\_ID <PK> | NUMBER | 123 |
| APPID | VARCHAR2 | Application ID |
| PRF\_STATE | VARCHAR2 | Refer Process Stages |
| FULL\_STATE | VARCHAR2 | Refer Process Stages |
| MAIN\_STATE | VARCHAR2 | Refer Process Stages |
| USERID | VARCHAR2 | User Name |
| REF\_ID | NUMBER | 789 |
| STATE\_DESC | VARCHAR2 | Refer Process Stages Description |
| \*START\_TIME | DATE | Timestamp |
| ^FINISH\_TIME | DATE | Timestamp |
| COMMENTS | VARCHAR2 | Comments from customer |
| OLP | CHAR | Y|N |
| OLP\_PRF\_FILE | CHAR | Y|N |
| OLP\_AUTO\_REL | CHAR | Y|N |
| OLP\_INT\_DLIST | VARCHAR2 | Email addresses |
| OLP\_DIVERT | CHAR | Y|N |
| TOTAL\_DIVERT\_PIECES | NUMBER | 0 |

“\*” – To be updated when state = ’submitted’

“^” – To be updated when state = ’failed/rejected/approved/disapproved/delete’

* Create **PROCESS\_MAIN\_EMT** table to keep state track of extended mailing jobs:
  + **TABLE: PROCESS\_MAIN\_EMT**

|  |  |  |
| --- | --- | --- |
| COLUMNS | TYPE | VALUE |
| PROC\_ID <PK> | NUMBER | 123 |
| STATE | VARCHAR2 | Submitted/ Warning/Success |
| PRF\_TXT\_NAME | VARCHAR2 | oscumlb11720181217160906.prf.txt |
| FULL\_TXT\_NAME | VARCHAR2 | osculc120181218081040.txt |
| WARNING\_CODE | NUMBER | 000 |
| WARNING\_DESC | VARCHAR2 | TBD FROM ISD |
| USERID | VARCHAR2 | User Name |
| \*START\_TIME | DATE | Timestamp |
| ^FINISH\_TIME | DATE | Timestamp |

“\*” – To be updated when state = ’submitted’

“^” – To be updated when state = ’success’

* ~~Create~~ **~~EMT\_ GLOBAL\_WARNING\_CODES~~** ~~table to keep maintain warning codes & description:~~
  + **~~TABLE: EMT\_GLOBAL\_WARNING\_CODES~~**

|  |  |  |
| --- | --- | --- |
| ~~COLUMNS~~ | ~~TYPE~~ | ~~VALUE~~ |
| ~~WARNING\_CODE <PK>~~ | ~~NUMBER~~ | ~~000/001/002 etc.~~ |
| ~~WARNING\_DESC~~ | ~~VARCHAR2~~ | ~~e.g. Missing PieceCode~~ |
| ~~LAST\_UPDATED~~ | ~~DATE~~ | ~~Timestamp~~ |

* Create a **PM\_EVENT\_LOG** table to keep track of various state changes:
  + **TABLE: PM\_EVENT\_LOG**

|  |  |  |
| --- | --- | --- |
| COLUMNS | TYPE | VALUE |
| ID | NUMBER | <SEQ\_NUMBER> |
| REF\_ID | NUMBER | <OLP\_ID>/<PROC\_ID> |
| TIMESTAMP | DATE | TIMESTAMP |
| SERVICE | VARCHAR2 | onlineProfing/extendedMailTracking |
| PRF\_STATE | VARCHAR2 |  |
| FULL\_STATE | VARCHAR2 |  |
| MAIN\_STATE | VARCHAR2 |  |
| MESSAGE | VARCHAR2 |  |
| USERID | VARCHAR2 |  |

Note: REF\_ID should not have any DB constraints. (E.g. no foreign keys to another table)

* **The following tables will be used & updated for ETA Board:**
  + **TABLE: IFS\_JOB\_LIST**

|  |  |  |
| --- | --- | --- |
| COLUMNS | TYPE | VALUE |
| JOB\_LIST\_ID | NUMBER |  |
| PROC\_ID | NUMBER |  |
| JOB\_NO | NUMBER |  |
| DT\_JOB\_ID | VARCHAR2 |  |
| APP\_NUM | NUMBER |  |
| CID | VARCHAR2 |  |
| APP\_NAME | VARCHAR2 |  |
| SITE | VARCHAR2 |  |
| JOB\_CREATION\_DATE | DATE |  |
| ETA | DATE |  |
| ARRIVAL\_TIME | DATE |  |
| DROP\_DATE | DATE |  |
| STATUS | NUMBER |  |
| DONE | NUMBER |  |

* + - get "JOB\_LIST\_ID" from ifs\_job\_list using proc\_id
  + **TABLE: IFS\_JOB\_DETAIL**

|  |  |  |
| --- | --- | --- |
| COLUMNS | TYPE | VALUE |
| JOB\_LIST\_ID | NUMBER |  |
| STATUS | NUMBER |  |
| MESSAGE | VARCHAR2 |  |
| TIME\_HOLD | DATE |  |
| REL\_MESSAGE | VARCHAR2 |  |
| TIME\_REL | DATE |  |
| USER\_ID\_HOLD | NUMBER |  |
| USER\_ID\_REL | NUMBER |  |

**Note:** Hold status is 2 & Release status is 3

**Process stages**:

|  |  |  |
| --- | --- | --- |
| **STATE** | **STATE\_DESCRIPTION** | **TEMPLATE\_ID** |
| Submitted | Job submitted to IndexFileProcessor | N/A |
| Warning | Job was not accepted/processed by IndexFileProcessor | jobStatusOLP.warning.notification |
| Success | Mongo DB file upload success | N/A |
| Failed | Process failed with fatal error | jobStatusOLP.failed |
| internal\_pending | Internal review/release pending | jobStatusOLP.readyForRelease |
| Released | Internal team released the job | jobStatusOLP.release |
| Rejected | Internal team rejected the job | jobStatusOLP.rejected |
| customer\_pending | Customer review/approval pending | jobStatusOLP.release |
| Approved | Customer approved the job | jobStatusOLP.approved |
| Disapproved | Customer disapproved the job | jobStatusOLP.disapproved |
| Delete | Job deleted | N/A |

**STATE columns with possible values:**

|  |  |  |
| --- | --- | --- |
| **PRF\_STATE** | **FULL\_STATE** | **MAIN\_STATE (Weight)** |
| Null | Null | null (0) |
| Warning | Warning | Warning (2) |
| Submitted | Submitted | submitted (1) |
| Success | Success | internal\_pending (2) / customer\_pending (3) |
| Failed | Failed | failed (5) |
|  |  | released (3) |
|  |  | rejected (5) |
|  |  | approved (4) |
|  |  | disapproved (5) |
|  |  | delete (5) |

**Folder hierarchy:**

* /var/www/InfoSync
  + InfoTrac
    - ExtendedMailTracking
    - OLP
    - Divert
  + EtaBoard
    - DashBoard
  + DataTrac

**InfoSync/InfoTrac/OLP/StateUpdate API Operations:**

* **[POST (INSERT)];**
  + 1. Check if a row exists in PROCESS\_MAIN table with the given PROC\_ID & CID If Yes go to step two else return HTTP 404 (PROC\_ID/CID Not Found)
  + 2. Check if OLP\_ID exist for given PROC\_ID & CID.
    - 2.1 If yes;
      * 2.1.1 Error: There’s an existing record for given PROD\_ID & CID, insert failed.
    - 2.2 If no;
      * 2.2.1 Insert a new row in the PROCESS\_MAIN\_OLP table using given parameters
      * 2.2.2 Update PROCESS\_MAIN table with OLP\_ID
      * 2.2.3 If PROCESS\_MAIN\_OLP:MAIN\_STATE value equals “submitted” OR “warning” OR “success” then
        + 2.2.3.1 Execute EMT\_STATE logic
      * 2.2.4 Insert a row in PM\_EVENT\_LOG
      * 2.2.5 Rest endpoint call returns
* **[PUT/PATCH (UPDATE)]**
  + 1. Check if a row exists in PROCESS\_MAIN table with the given PROC\_ID & CID If Yes go to step two else return HTTP 404 (PROC\_ID/CID Not Found)
  + 2. Check if OLP\_ID exist for given PROC\_ID & CID.
    - 2.1 If yes;
      * 2.1.1 Extract OLP\_ID using given PROC\_ID & CID
      * 2.1.2 Update PROCESS\_MAIN\_OLP table using OLP\_ID with given parameters [Refer Pseudo Code Logic]
      * 2.1.3 If PROCESS\_MAIN\_OLP:MAIN\_STATE value equals “submitted” OR “warning” OR “success” then
        + 2.1.3.1 Execute EMT\_STATE logic
      * 2.1.4 Insert a row in PM\_EVENT\_LOG
      * 2.1.5 Rest endpoint call returns
    - 2.2 If no;
      * 2.2.1 Error: There’s no existing OLP record to update for given PROD\_ID & CID

**InfoSync/InfoTrac/OLP/StateUpdate API parameters:**

\*proc\_id: Unique Id

\*cid

\*state:

\*userId:

state\_desc:

olp:

olp\_prf\_file

olp\_auto\_rel

olp\_int\_dlist

olp\_divert

message

warning\_code

warning\_desc

prf\_txt\_name

full\_txt\_name

**\*required**

**InfoSync/InfoTrac/OLP/GetState API Operations:**

* 1. GetState is going to be a batch call containing one cid and one/multiple PROC\_ID separated by ‘,’.
* 2. For each CID/PROC\_ID check if a row exists in PROCESS\_MAIN table If Yes retrieve the OLP\_ID & go to step three else construct the error code for given cid/proc\_id.
* 3. Retrieve the following values from PROCESS\_MAIN\_OLP table based on OLP\_ID.
  + olp = “$olp”
  + olp\_auto\_rel = “$olp\_auto\_rel”
  + olp\_divert = “$olp\_divert”
  + olp\_id = $olp\_id
  + olp\_int\_dlist = “$olp\_int\_dlist”
  + olp\_prf\_file = “$olp\_prf\_file”
  + proc\_id = “$proc\_id”
  + start\_time=”<timestamp>”
  + state = “$main\_state”
  + success= true/false
  + total\_divert\_pieces=$total\_divert\_pieces
  + userid = “$userid”
* 4. Respond back with JSON object.

**InfoSync/InfoTrac/OLP/GetState API parameters:**

\*proc\_id: Unique Id

\*cid

**\*required**

**InfoSync/InfoTrac/ExtendedMailTracking/JobTracking API Operations:**

* + The JobTracking API will be developed under ExtendedMailTracking
    - * If STATE value equals “submitted” OR “warning” OR “success” then Execute EMT\_STATE logic

**InfoSync/InfoTrac/ExtendedMailTracking/JobTracking API parameters:**

\*proc\_id: Unique Id

\*cid

\*state

\*userId

message

warning\_code

warning\_desc

prf\_txt\_name

full\_txt\_name

**\*required**

**InfoSync/InfoTrac/ExtendedMailTracking/GetJobState Operations:**

* 1. Query PROCESS\_MAIN\_EMT based on given parameters
* 2. Check if Status column = 1 in PROCESS\_MAIN for each record from step 1:
  + 2.1 If yes;
    - 2.1.1 Include the record in response JSON with OLP\_ID value
  + 2.2 If no;
    - 2.2.1 Do not include the record in response JSON.
* 3. Return response JSON with below values:
  + - Proc\_id
    - cid
    - warning\_code
    - warning\_desc
    - date\_range:
    - olp\_id
    - prf\_txt\_name
    - full\_txt\_name

**InfoSync/InfoTrac/ExtendedMailTracking/GetJobState API parameters:**

\*State: submitted, warning, success

**archivalPeriod**

**\*required**

**Note – archivalPeriod default value should be set to 90 days**

**InfoSync/EtaBoard/DashBoard/UpdateJobStatus API Operations:**

* 1. Check if a row exists in IFS\_JOB\_LIST table with the given PROC\_ID If Yes go to step two else return HTTP 404 (PROC ID Not Found)
* 2. Check if a row with extracted JOB\_LIST\_ID is present in IFS\_JOB\_DETAIL table If yes go to step three else return HTTP 404 (JOB\_LIST\_ID Not Found in IFS\_JOB\_DETAIL)
* 3. Update the IFS\_JOB\_DETAIL row with passed parameters
  + When status = 2 i.e. hold
    - auto generate the timestamp for time\_hold column
    - update IFS\_JOB\_LIST record to set status = 2
  + When status = 3 i.e. release
    - auto generate the timestamp for time\_rel column
* 4. Rest endpoint call returns

**InfoSync/EtaBoard/DashBoard/UpdateJobStatus API parameters:**

\*proc\_id: Unique Id

\*status:

message:

rel\_message:

user\_id\_hold:

user\_id\_rel:

**\*required**

* **Passing parameters based on state:**
  + **if main\_State = “submit”**
    - \*proc\_id: Unique Id
    - \*status: 2
    - message: “Job submitted to IndexFileProcessor”
    - user\_id\_hold: $user\_id
  + **if main\_State = “approved”**
    - \*proc\_id: Unique Id
    - \*status: 3
    - rel\_message: “Job is approved by the customer”
    - user\_id\_rel: SysOper

**Pseudo Code Logic (To calculate Process\_main\_olp:Main\_State value):**

* Find out passed.state.weight based on mapping table
* Get the current.main\_state.weight
* Check if [type] parameter is present
  + If yes: (The call is for either prf or full state)
    - If current.main\_state.weight < passed.state.weight
      * If passed.state value eq “success”
        + find out the main\_state value based on olp\_auto\_rel value

if olp\_auto\_rel = ‘Y’ then main\_state = ‘customer\_pending’

if olp\_auto\_rel = ‘N’ then main\_state = ‘internal\_pending’

* + - * + Update prf/full state column w/ ‘success’ & main state column w/ customer\_pending or internal\_pending
      * else
        + Update prf/full state column & main state column w/ given state
    - Else
      * If current.main\_state.weight < 5
        + Then update prf/full state column w/ given state
      * Else
        + Error – The current state for [type] cannot be updated from current.state to passed.state as main state is set to current.main\_state
  + If no: (The call is for main state)
    - If current.main\_state.weight < passed.state.weight
      * Then update main state column w/ given state
    - Else
      * Error – The current main state cannot be updated from current.main.state to passed.main.state

**EMT STATE Logic**:

* 1. Check if a record exists with given proc\_Id
  + 1.1 If yes;
    - 1.1.1 Update the record with given parameters ONLY if passed state value does not equal to current process\_main\_emt:state value
  + 1.2 If no;
    - 1.2.1 Insert a new record with given parameters
  + 1.3 Create a row in PM\_EVENT\_LOG table with service value “extendedMailTracking”

MongoDB

Process Print TXT Files

Proces

IndexFileProcessor

Oracle

[PROC\_ID]

[CID]

[STATE]

[USER\_ID]

[TYPE]

Job Status

Query OLP Jobs

Proces

InfoServices API

[{proc\_id, response, state}, {proc\_id, response, state}, {…}]

{email\_dist, template\_id, content\_data.json}

[{proc\_id, cid}, {proc\_id, cid}, {…}]

internal\_pending/ customer\_pending/ failed

submitted

InfoSync API

Trigger Idx File Processor API

Proces

Print Platform

Step 1

Step 2

Step 3

delete

internal\_released / internal\_rejected

ETABoard API

completed

Oracle

Oracle

Internal OLP Release To Customer

Proces

Internal OLP Release To Customer

Proces

InfoServices API

[PROC\_ID]

[CID]

[STATE]

[USER\_ID]

[PROC\_ID]

[CID]

[STATE]

[USER\_ID]

Oracle

Oracle

Update State

Proces

Update State

Proces

InfoSync API

Print Platform

Step 4B

Step 4A

approved / disapproved

Update ETA Approval Status

Proces

ETABoard API

Oracle

Internal OLP Release To Customer

Proces

InfoServices API

[PROC\_ID]

[CID]

[STATE]

[USER\_ID]

[INFOTRAC\_LOGIN\_ID

Update State

Proces

Oracle

InfoSync API

Step 4C

Print Platform

**Step 1:**

Description:

Print processing platform makes a call to IndexFileProcessor API to process the print TXT files.

Requirement #1:

* TXT Index File Structure Updates needed. See red font below for the new fields:

File Naming Convention:

/d/isd\_share/[CID]/print\_fileid/[cid][jid]YYYYMMDDHHMMSS

afcums120170901144852.txt

[HEADER]

hostname=rhs

user=pdev1@rhstore

proc\_id=1369852

cid=afcu

jid=ms1

job\_2d = <3digit code>

fileid\_path=/d/isd\_share/afcu/print\_fileid/

afp\_path=/d/isd\_share/afcu/afpfiles/

mid=901141739

service\_id=041

track\_by\_rec=y

imb\_track=y

olp=y

olp\_divert=y

[IDX\_FILE]

afcums101e.sup.20170901144852

afcums1015.nop.20170901144852

afcums1014.2o1.20170901144852

afcums1015.2oz.20170901144852

afcums1015.rtm.20170901144852

afcums1015.fgn.20170901144852

afcums1015.hvy.20170901144852

[AFP FILE]

afcums1014.2o1.afp.20170901144852

afcums1015.2oz.afp.20170901144852

afcums1015.rtm.afp.20170901144852

afcums1015.fgn.afp.20170901144852

afcums1015.hvy.afp.20170901144852

EOF

The following are the explanation of the additional new arguments

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Arguments** | **Description** | **Dependencies** |
| Job\_2d | <3 digit code> | 3 digit job id from print side | None |
| Olp | y: yes  n: no | When this field is present and the value set to ‘y’, then this job requires online proofing.  If this field is not present OR if the value is ‘n’, there is no Online Proofing needed | None; Optional Field |
| olp\_prf\_file | y: yes  n: no | If value set to ‘y’, [the proof file with specified accounts needs to be processed]  If this field is not present OR if the value is ‘n’, [Do not process the proof file] | olp |
| olp\_auto\_rel | y: yes  n: no | If value set to ‘n’, then this job requires internal review before making this file available to client. If this field is not present OR if the value is ‘y’, there is no need for internal review. Immediately inform customer to review the file. | olp |
| Olp\_int\_dlist | <email ids>  null | List of internal email addresses that need to be notified; if not defined the default email id is: “dp@infoimageinc.com” | olp; olp\_auto\_rel |
| olp\_divert | y: yes  n: no | If value set to ‘n’, [the job does not have a divert feature]. If this field is not present OR if the value is ‘y’, [divert feature is enabled] | olp |

**Step 2:**

After receiving the request from the Index File Index file processor, we will need to make another API call to InfoServiceAPI to trigger email notification depending on the states. Refer to the “Process Stages” section of the documentation for the listing of all the states.

The following are the http payload parameters that will be submitted to the InfoServices API.

URL: [PENDING FROM ANIL/DAVID T.]

HTTP POST PARAMS:

|  |  |
| --- | --- |
| cid | InfoIMAGE’s 4 character client id; e.g. cid = “afcu” |
| infotrac\_feature | InfoTRAC UI’s feature name. This will used by the InfoTRAC to lookup the feature business rule that may be required for interpreting / parsing the email data that we are submitting.  [ANIL/DAVID T. WILL PROVIDE US THE EXACT FEATURE VALUE FOR THIS ONLINE PROOFING FEATURE] |
| content\_data.json | This is a JSON string containing the email content key/value data, template\_id to use and the optional email\_list that we may or may not provide. Refer to the subsequent section to describe how this used and the specifications per template\_id. |

The following are the http payload parameters that will be submitted to the InfoServices API.

URL: [PENDING FROM ANIL/DAVID T.]

HTTP POST PARAMS:

|  |  |
| --- | --- |
| cid | InfoIMAGE’s 4 character client id; e.g. cid = “afcu” |
| infotrac\_feature | InfoTRAC UI’s feature name. This will used by the InfoTRAC to lookup the feature business rule that may be required for interpreting / parsing the email data that we are submitting.  [ANIL/DAVID T. WILL PROVIDE US THE EXACT FEATURE VALUE FOR THIS ONLINE PROOFING FEATURE] |
| content\_data.json | This is a JSON string containing the email content key/value data, template\_id to use and the optional email\_list that we may or may not provide. Refer to the subsequent section to describe how this used and the specifications per template\_id. |

**Template ID**: jobStatusOlp.readyForRelease

**[EMAIL TEMPLATE]**

**from:**

Data.Center@infoimageinc.com

**subject:**

${cid} - ${application\_id}: Online Proofing – Internal Review Request

**body:**

Dear InfoIMAGE Team,  
  
Your ${application\_id} online proofing documents have been posted into InfoTRAC as of ${create\_time}. Please logon to the InfoTrac system and select the Job Status navigation menu to review Online Proofing documents and click on the release icon if you feel that it is ready for the client to review.  
  
Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388  
Data.Center@infoimageinc.com

Ref#: ${dt\_job\_id}{$proc\_id}

**[JSON DATA]**

“template\_id”: “jobStatusOlp.readyForRelease”,

“email\_list”: [{ac@infoimageinc.com}, {dp@infoimageinc.com}]

“email\_variables”:

{

“cid”, “afcu”,

“application\_id”: ”DDA\_Statement”,

“create\_time”: ””,

“dt\_job\_id”: “141574-C03.01.V22204”,

“proc\_id”:12345

},

NOTE:

ac@infoimageinc.com will be used for the Tax Applications.

dp@infoimageinc.com will be used for the other applications outside of Taxes.

**Template ID**: jobStatusOlp.release

**[EMAIL TEMPLATE]**

**from:**

Data.Center@infoimageinc.com

**subject:**

${cid}- ${application\_id}: Online Proofing - New Sample Available

**body:**

Dear Valued Customers,  
  
The ${application\_id} online proofing documents are ready for review as of ${create\_time}. Please logon to the InfoTrac system and select the Job Status navigation menu to review and approve.  
  
Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388  
Data.Center@infoimageinc.com

Ref#: ${dt\_job\_id}{$proc\_id}

**[JSON DATA]**

“template\_id”: “jobStatusOlp.release”,

“email\_list”: [{}]

“email\_variables”:

{

“cid”, “afcu”,

“application\_id”: ”DDA\_Statement”,

“create\_time”: ””,

“dt\_job\_id”: “141574-C03.01.V22204”,

“proc\_id”:12345

},

**Template ID**: jobStatusOlp.approved

**[EMAIL TEMPLATE] –**

**from:**

Data.Center@infoimageinc.com

**subject:**

${cid} - ${application\_id}: Online Proofing - Sample Approved

**body:**

Dear Valued Customer,  
  
${infotrac\_login\_id} have approved the ${application\_id} samples as of ${approved\_time} with the following production volume counts:  
  
Sample Creation Date: ${create\_time}  
# of Printed Accounts: ${total\_printed}  
# of Suppressed Accounts: ${total\_non\_print}  
# of Print Suppression by eStatement Enrollment: ${total\_esup}  
# of Divert Pieces: ${ total\_divert\_pieces }

Total Account Volume: ${grand\_total}

Your Comments:  
==========================================================================  
  
${comments}  
  
==========================================================================  
  
  
Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388  
Data.Center@infoimageinc.com

Ref#: ${dt\_job\_id}{$proc\_id}

**[JSON DATA]**

“template\_id”: “jobStatusOlp.approved”,

“email\_list”: [{}]

“email\_variables”:

{

“cid”, “afcu”,

“application\_id”: ”DDA\_Statement”,

“infotrac\_login\_id:”${infotrac\_login\_id}”,

“approved\_time”: ””,

“create\_time”: ””,

“total\_printed”: ””,

“total\_non\_print”: ”0”,

“total\_esup”: “”,

“total\_divert\_pieces “:”${ total\_divert\_pieces}”,

“grand\_total”: “11”,

“comments”: “”,

“dt\_job\_id”: “141574-C03.01.V22204”,

“proc\_id”:12345

},

**Template ID**: jobStatusOlp.disapproved

**[EMAIL TEMPLATE]**

**from:**

Data.Center@infoimageinc.com

**subject:**

${cid} - ${application\_id}: Online Proofing - Sample Not Approved

**body:**

Dear Valued Customer,  
  
${infotrac\_login\_id} has selected to **not** approve the ${application\_id} samples as of ${approved\_time}. Please contact your Account Coordinator to review the issue/concerns regarding these samples.

Your Account Coordinator is:  
${{client.accountCoordinator.fullName}}  
Account Coordinator  
InfoIMAGE Inc.  
${{client.accountCoordinator.phone}}  
${{client.accountCoordinator.email}}

Here were the comments from ${infotrac\_login\_id}:

Your Comments:  
==========================================================================  
  
${comments}  
  
==========================================================================  
  
  
Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388  
Data.Center @infoimageinc.com

Ref#: ${dt\_job\_id}{$proc\_id}

**[JSON DATA]**

“template\_id”: “jobStatusOlp. disapproved”,

“email\_list”: [{}]

“email\_variables”:

{

“cid”, “afcu”,

“application\_id”: ”DDA\_Statement”,

“infotrac\_login\_id:”${infotrac\_login\_id}”,

“approved\_time”: ””,

“comments”: “”,

“client\_accountCoordinator\_fullName”: “${client.accountCoordinator.fullName}”,

“client\_account Coordinator\_phone”:”${client.accountCoordinator.phone}”,

“client\_accountCoordinator\_email”:”${client.accountCoordinator.email}”

“dt\_job\_id”: “141574-C03.01.V22204”,

“proc\_id”:12345

},

**Template ID**: jobStatusOlp.failed

**[EMAIL TEMPLATE]**

**from:**

Data.Center@infoimageinc.com

**subject:**

${cid} - ${application\_id}: Online Proofing – Processing Failed

**body:**

Dear InfoIMAGE Team,  
  
The ${application\_id} online proofing processing failed.

DT\_JOB\_ID: ${dt\_job\_id}

PROC\_ID: {$proc\_id}

Error Message: {$error\_message}

Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388  
Data.Center@infoimageinc.com

Ref#: ${dt\_job\_id}{$proc\_id}

**[JSON DATA]**

“template\_id”: “jobStatusOlp.failed”,

“email\_list”: [{“email”: “dp@infoimageinc.com”}]

“email\_variables”:

{

“cid”, “afcu”,

“application\_id”: ”DDA\_Statement”,

~~“create\_time”: ””,~~

“error\_message”: ””,

“dt\_job\_id”: “141574-C03.01.V22204”,

“proc\_id”:12345

},

**Template ID**: jobStatusOlp.rejected

**[EMAIL TEMPLATE]**

**from:**

Data.Center@infoimageinc.com

**subject:**

${cid} - ${application\_id}: Online Proofing – Not Released To Customer

**body:**

Dear InfoIMAGE Team,  
  
${infotrac\_login\_id} has selected to **not** to release the Online Proofing file for ${application\_id}.

Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388  
Data.Center @infoimageinc.com

Ref#: ${dt\_job\_id}{$proc\_id}

**[JSON DATA]**

“template\_id”: “jobStatusOlp.rejected”,

“email\_list”: [{“email”: “dp@infoimageinc.com”}]

“email\_variables”:

{

“cid”, “afcu”,

“application\_id”: ”DDA\_Statement”,

“infotrac\_login\_id:”${infotrac\_login\_id}”,

“dt\_job\_id”: “141574-C03.01.V22204”,

“proc\_id”:12345

},

**Template ID**:  jobStatusOLP.warning.notification

**[EMAIL TEMPLATE]**

**from:**

[Data.Center@infoimageinc.com](mailto:Data.Center@infoimageinc.com)

**subject:**

${cid}${jid} - ${job\_id}: Warning message

**body:**

Dear InfoIMAGE Team,  
  
Please take corrective actions to resolve following warning message.

DT\_JOB\_ID:  ${dt\_job\_id}

PROC\_ID: ${proc\_id}

Error Message: ${error\_message}

Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388  
[Data.Center@infoimageinc.com](mailto:Data.Center@infoimageinc.com)

Ref#:  ${dt\_job\_id}{$proc\_id}

**[JSON DATA]**

“template\_id”: “jobStatusOLP.warning.notification”,

“email\_list”: [{ac@infoimageinc.com}, [{dp@infoimageinc.com}](mailto:%7bdp@infoimageinc.com%7d)],

“email\_variables”:

                {

“cid”, “afcu”,

“jid”, “ms1”,

                                “job\_id”: ”DDA\_Statement”,

                                “error\_message”: ””,

                                “dt\_job\_id”: “141574-C03.01.V22204”,

                                “proc\_id”:”12345”

                }

**Template ID**:  jobStatus.PieceCode.notification - TBD

**[EMAIL TEMPLATE]**

**~~from:~~**

[~~Data.Center@infoimageinc.com~~](mailto:Data.Center@infoimageinc.com)

**~~subject:~~**

~~${cid}${jid} - ${job\_id}: Warning message~~

**~~body:~~**

~~Dear InfoIMAGE Team,  
  
Please take corrective actions to resolve following warning message.~~

~~DT\_JOB\_ID:  ${dt\_job\_id}~~

~~PROC\_ID: ${proc\_id}~~

~~Error Message: ${error\_message}~~

~~Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388~~[~~Data.Center@infoimageinc.com~~](mailto:Data.Center@infoimageinc.com)

~~Ref#:  ${dt\_job\_id}{$proc\_id}~~

**~~[JSON DATA]~~**

~~“template\_id”: “jobStatusOLP.warning.notification”,~~

~~“email\_list”: [{ac@infoimageinc.com},~~ [~~{dp@infoimageinc.com}~~](mailto:%7bdp@infoimageinc.com%7d)~~],~~

~~“email\_variables”:~~

~~{~~

~~“cid”, “afcu”,~~

~~“jid”, “ms1”,~~

~~“job\_id”: ”DDA\_Statement”,~~

~~“error\_message”: ””,~~

~~“dt\_job\_id”: “141574-C03.01.V22204”,~~

~~“proc\_id”:”12345”~~

~~}~~

**Template ID**:  jobStatus.PendingJobs.notification - TBD

**[EMAIL TEMPLATE]**

**~~from:~~**

[~~Data.Center@infoimageinc.com~~](mailto:Data.Center@infoimageinc.com)

**~~subject:~~**

~~${cid}${jid} - ${job\_id}: Warning message~~

**~~body:~~**

~~Dear InfoIMAGE Team,  
  
Please take corrective actions to resolve following warning message.~~

~~DT\_JOB\_ID:  ${dt\_job\_id}~~

~~PROC\_ID: ${proc\_id}~~

~~Error Message: ${error\_message}~~

~~Thank you,  
  
Data Center Department  
InfoIMAGE Inc.  
650.473.6388~~[~~Data.Center@infoimageinc.com~~](mailto:Data.Center@infoimageinc.com)

~~Ref#:  ${dt\_job\_id}{$proc\_id}~~

**~~[JSON DATA]~~**

~~“template\_id”: “jobStatusOLP.warning.notification”,~~

~~“email\_list”: [{ac@infoimageinc.com},~~ [~~{dp@infoimageinc.com}~~](mailto:%7bdp@infoimageinc.com%7d)~~],~~

~~“email\_variables”:~~

~~{~~

~~“cid”, “afcu”,~~

~~“jid”, “ms1”,~~

~~“job\_id”: ”DDA\_Statement”,~~

~~“error\_message”: ””,~~

~~“dt\_job\_id”: “141574-C03.01.V22204”,~~

~~“proc\_id”:”12345”~~

~~}~~

**Job Monitoring Script**

**Develop a monitoring script that will invoke GetExtendedMailJobState API to get the list of jobs that are pending for PieceCode update based on warning\_code. Analyze the response & report pending jobs to PC twice a day (Time - TBD) using given email template (jobStatus.PieceCode.notification).**

**Email JSON Variable Logic**

Here’s the logic to some to get the values from each of these specific variables:

application\_id:

-- Following query is using Microsoft SQL Server. (10.8.8.28)

-- the vid value is from the DT\_JOB\_ID string.

-- e.g. 136385-C27.01-V**20672**

-- 20672 is the version id number (the number after the letter ‘V’)

SELECT APPID AS application\_id  
FROM tblAl  
WHERE cid = UPPER(<cid>)  
AND vid =<vid>  
AND active = 'true'

SELECT DT\_JOB\_ID FROM PROCESS\_MAIN

WHERE cid = <cid> AND proc\_id = <proc\_id>;

grand\_total:

~~SELECT SUM(acct\_cnt)~~

~~FROM dpvs\_cnt~~

~~WHERE proc\_id = <proc\_id>;~~

**SELECT SUM(acct\_cnt)**

**FROM dpvs\_cnt**

**WHERE proc\_id = <proc\_id>**

**AND file\_id not in ('pdf1','pdf2','pdf3','5.rpt',**

**'5.np1','5.np2','5.np3','5.npm');**

total\_esup:

SELECT sum(acct\_cnt)

FROM dpvs\_cnt

WHERE proc\_id = <proc\_id>

AND file\_id in ('e.sup', 'e.sup2');

**NOTE: Any dpvs\_cnt tabled related query should be stored in a class. This class may be re-used for other internal coding for our APIs. If you don’t remember to do this now, I will add this as an Phase II item to develop this later.**

create\_time:

SELECT pmo.start

FROM process\_main pm, process\_main\_olp pmo

WHERE pm.proc\_id = <proc\_id>

AND pm.olp\_id = pmo.olp\_id

approved\_time:

SELECT pmo.finish\_time

FROM process\_main pm, process\_main\_olp pmo

WHERE pm.proc\_id = <proc\_id>

AND pm.olp\_id = pmo.olp\_id

AND pmo.main\_state = ‘approved’ OR pmo.main\_state = ‘disapproved’

error\_message:

-- The following query may return more than one result use. Use only the first record that it returns.

SELECT pel.message  
FROM process\_main pm, process\_main\_olp pm\_olp, pm\_event\_log pel  
WHERE pm.proc\_id = <proc\_id>  
AND pm.olp\_id = pm\_olp.olp\_id  
AND pm\_olp.ref\_id = pel.ref\_id  
AND pel.main\_state = ‘FAILED’  
ORDER BY pel.id desc;

infotrac\_login\_id:

This value is included within the JSON body of the PATCH method. When InfoTRAC approve or disapprove the online proofing job, the InfoServices API will include the infotrac login id name when submitting the PATCH update API

total\_printed:

**SELECT SUM(acct\_cnt)**

**FROM dpvs\_cnt**

**WHERE proc\_id = <proc\_id>**

**AND file\_id not in ('e.sup','e.sup2','nonEstmt','NONPRINT',**

**'pdf1','pdf2','pdf3','5.rpt',**

**'5.np1','5.np2','5.np3','5.npm');**

total\_non\_print:

**SELECT SUM(acct\_cnt)**

**FROM dpvs\_cnt**

**WHERE proc\_id = <proc\_id>**

**AND file\_id in ('e.sup','e.sup2','nonEstmt','NONPRINT');**

comments:

SELECT olp.comments

FROM process\_main\_olp olp, process\_main pm

WHERE pm.proc\_id = <proc\_id>

AND pm.cid = <cid>

AND pm.olp\_id = olp.olp\_id;

**Step 4 A, B, & C:**

|  |  |
| --- | --- |
| **Weight** | **State(s)** |
| 0 | Null |
| 1 | Submitted |
| 2 | warning/success/internal\_pending |
| 3 | released / customer\_pending |
| 4 | Approved |
| 5 | failed/rejected/disapproved/delete |

Below is the list of possible MAIN\_STATE values:

|  |
| --- |
| **Response** |
| NULL |
| WARNING |
| SUBMITTED |
| FAILED |
| INTERNAL\_PENDING |
| CUSTOMER\_PENDING |
| RELEASED |
| REJECTED |
| APPROVED |
| DISAPPROVED |
| DELETE |

**URL:**

<http://is-apps-dev.infoimage.com:9090> - **Dev**

<http://is-apps-dev.infoimage.com> - **Pilot**

<http://is-apps-stage.infoimage.com> - **Stage**

<http://is-apps-prod.infoimage.com> - **Prod**

**FILE ID CONVERSION TABLE**

Please refer “File\_ID\_Standard” table for complete list of File IDs. E.g.

|  |  |
| --- | --- |
| File ID | 2digit Camera File ID |
| 4.2o1 – 4.2oq | 4a-4z |
| 5.2oz | 5a |
| 5.fgn | 5f |
| 5.hvy | 5h |
| 5.mis | 5m |
| 5.pm1 – 5.pm5 | p1-p5 |

**Open questions:**

* ~~Confirm the prf file format; afp or pdf?~~ – **PDF (Confirmed by Peter 9/22)**
* ~~Are we going to copy non-print files to PD side?~~ **If there is time to do so, yes; If not, phase 2. (Note: If this is implemented, please check the behavior as to whether or not the IndexFileProcessor API will process the non-print files. We want to make sure the code ingests this AFP file).**
* ~~10 Digit unique id will be added to txt or PD needs to construct?~~ – **PD needs to construct; IS will provide job\_2d. 10 Digit example – ABC-4a-00001 (jobId-fileId-seqNum)**
* Request an update after certain time\* or wait for the response back
* Provide time stats from current index file processor - **ROHIT**
* Would adding new fields to index file affect the existing processing? - **ROHIT**
* DSI database access.
* VM for Python 3? Dockers can be used as VM alternative
* ~~Do we need to design Piece\_Trac table or an existing combination of tables cab be used?~~ – **Existing set of tables can be leveraged.**
* Job Status Tab tuning - **ROHIT**
* ~~Mongo DB failure capture?~~ – **n/a at this time; not using MongoDB; Subject to change when we discuss the divert portion of the project.**

**Important:**

* All ‘olp’ flag & ‘mail\_full\_serv‘/’track\_by\_rec‘/’fullimb\_trk‘ validation should be done at the beginning of the process.
* For configuration use system.properties files
* Phase II – Add ASE 3 & ASE4 condition

**Reference Docs:**

Swagger KT document -> “/d/iss/shail/2017-08-09 15.08 IS Training - IS Web App \_ Trigger System.mp4”

Python code for DB connection -> afp2pdf\_report\_handler.py

Refer Document template on EasyRedMine

**Phase – II:**

* Control all the flags using on/off button on UI
* Use python script to convert from afp to pdf
* PDFI approval notification workflow
* PDF link failure due to missing account number
* Log file updates

**Release Notes:**

DDL for the REST end point operations:

**CREATE TABLE PROCESS\_MAIN\_OLP (**

**OLP\_ID      NUMBER        NOT NULL PRIMARY KEY,**

**PRF\_STATE   VARCHAR2(200),**

**FULL\_STATE  VARCHAR2(200),**

**USERID      VARCHAR2(200),**

**STATE\_DESC  VARCHAR2(200),**

**START\_TIME     DATE,**

**FINISH\_TIME      DATE,**

**REF\_ID      NUMBER,**

**MAIN\_STATE VARCHAR(200),**

**OLP CHAR(1),**

**OLP\_PRF\_FILE CHAR(1),**

**OLP\_AUTO\_REL CHAR(1),**

**OLP\_INT\_DLSIT VARCHAR2(4000),**

**OLP\_DIVERT CHAR(1),**

**COMMENTS VARCHAR2(4000)**

**);**

**CREATE TABLE PM\_EVENT\_LOG**

**(**

**ID            NUMBER        NOT NULL PRIMARY KEY,**

**REF\_ID        NUMBER        NOT NULL,**

**TIMESTAMP     DATE          NOT NULL,**

**SERVICE       VARCHAR2(200) NOT NULL,**

**MESSAGE       VARCHAR2(4000),**

**USERID VARCHAR2(200),**

**PRF\_STATE VARCHAR2(200),**

**FULL\_STATE VARCHAR2(200),**

**MAIN\_STATE VARCHAR2(200)**

**);**

**ALTER TABLE PROCESS\_MAIN ADD OLP\_ID NUMBER NULL;**

**CREATE SEQUENCE PMEL\_AUTO\_INCR MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 2701 CACHE 20 NOORDER NOCYCLE;**

**CREATE SEQUENCE PMO\_AUTO\_INCR MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1301 CACHE 20 NOORDER NOCYCLE ;**

**Work with Network team to:**

* Enable site monitoring of our API services.  Provide them one API GET service call and the example JSON response body that the monitoring system will parse and ensure it matches.

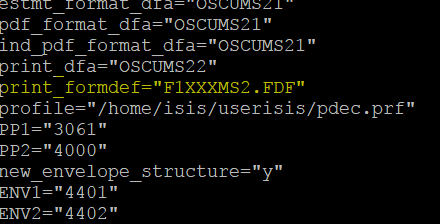
**PHASE – II release notes:**

* Ensure all the URLs are pointing to correct env i.e. dev/pilot/stage/prod
* Update email\_list in the config file per env

**FormDef – Change Specs**

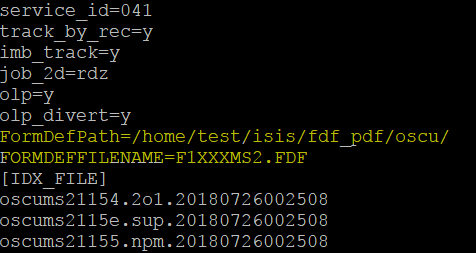
**Process/Tasks:**

- Print side to Include a flag in a control file: e.g.



-Update isisdisk.sh & isisdisk\_daily.sh to read print\_formdef flag & write it to e-stmt index e.g.

*/z/isd\_share/oscu/print\_fileid/oscums220180726002508.txt*



*-Note: The change is applicable to all the applications regardless they have AOLP or not. E-stmt index should include the Formdef flags only when Formdef flag is present in a control file.*

-Modify check in scripts to include/accept **ppfa & fdf\_pdf** folders.

-ISD will modify their afp2pdf call to point preprocessing03 & include required parameters.

**Command Line Changes:**

**Pilot (Existing call on preprocessing02)**

/home/afp2web/afp2web -ALL -so -op:/d/isd\_share/afp2web\_out -pp:1-3 -if:/home/keep/afp2web\_parallel.ini -q /z/isd\_share/oscu/afpfiles/oscums21155.2oz.afp.20180726002508 >/d/isd\_share/afp2web\_out/59204f27-1896-4739-ba5f-2bcea86ee158.pdf

**Pilot (Proposed preprocessing03 call)**

python /home/test/master/afp2pdf.py direct FileCreationMode=ALL StartingPage=1 EndingPage=3 IniFilePath=/home/test/keep/ IniFilename=afp2web\_pr.ini InputFilename=/z/isd\_share/oscu/afpfiles/ oscums21155.2oz.afp.20180726002508 OutputFilePath=/d/isd\_share/afp2web\_out/ FormDefPath=/home/test/isis/fdf\_pdf/oscu/ FORMDEFFILENAME=F1XXXMS2.FDF Quietmode=on FilenamePattern="<59204f27-1896-4739-ba5f-2bcea86ee158>"

**Stage (Existing Call on preprocessing02)**

/home/afp2web/afp2web -ALL -so -op:/d/isd\_share/afp2web\_out -pp:1-3 -if:/home/keep/afp2web\_parallel.ini -q /z/isd\_share/oscu/afpfiles/oscums21155.2oz.afp.20180726002508 >/d/isd\_share/afp2web\_out/ca33e9f9-f2e9-4d08-9d34-e60a41e0bc77.pdf

**Stage (Proposed preprocessing03 call)**

python /home/test/master/afp2pdf.py direct FileCreationMode= ALL StartingPage=1 EndingPage=3 IniFilePath=/home/test/keep/ IniFilename=afp2web\_pr.ini InputFilename=/z/isd\_share/oscu/afpfiles/ oscums21155.2oz.afp.20180726002508 OutputFilePath=/d/isd\_share/afp2web\_out/ FormDefPath=/home/test/isis/fdf\_pdf/oscu/ FORMDEFFILENAME= F1XXXMS2.FDF Quietmode=on FilenamePattern="< ca33e9f9-f2e9-4d08-9d34-e60a41e0bc77 >"

**Production (Existing Call on preprocessing02)**

/home/afp2web/afp2web -ALL -so -op:/d/isd\_share/afp2web\_out -pp:38-38 -if:/home/keep/afp2web.ini -q /d/isd\_share/alcu/afpfiles/alcudla1254.2o1.afp.20180926060338 >/d/isd\_share/afp2web\_out/df1d6189-0a05-40fb-ab28-d6611990a3ca.pdf

**Production (Proposed preprocessing03 call)**

python /home/test/master/afp2pdf.py direct FileCreationMode= ALL StartingPage=38 EndingPage=38 IniFilePath=/home/test/keep/ IniFilename=afp2web\_prod.ini InputFilename=/d/isd\_share/alcu/afpfiles/alcudla1254.2o1.afp.20180926060338 OutputFilePath=/d/isd\_share/afp2web\_out/ FormDefPath=/home/test/isis/fdf\_pdf/alcu/ FORMDEFFILENAME=F1XXXMS1.FDF Quietmode=on FilenamePattern="< df1d6189-0a05-40fb-ab28-d6611990a3ca>"

**Formdef paths (OSCU MS2 example):**

/home/isis/ppfa/oscu/

/home/isis/fdf\_pdf/oscu/

/home/isis/ppfa\_s\_src/oscu/ms2/ogl/

/home/isis/ppfa\_s\_src/oscu/ms2/ovl300/

/home/isis/ppfa\_s\_src/oscu/ms2/tiff/

/home/isis/ppfa\_s\_src/oscu/ms2/pseg/

/home/isis/ppfa\_f\_src/ (Not available yet – E-statements resources)

/home/isis/ppfa\_e\_src/ (Not available yet – Archival resources)

**Naming conventions (OSCU MS2 example):**

PPFA - XXXMS2.PFA

FORMDEF - F1XXXMS2.FDF

OGL – XXXXXX.OGL - TBD

OVL – O1XXXXXX.OVL - TBD

TIFF – OSCUXXX.TIF