



ADVANCE ONLINE PROOFING

User Reference Guide

Revision v0.1

Table of Contents:

Synopsis:	3
Design Description:	3
Implementation workflow:	4
Application Logic - Flowchart:.....	4
IS Database Schema changes:	5
Process Stages / E-mail templates:	8
Environment & Folder Hierarchy:	8
API specs & operations:	9
Processing Steps:.....	14
Divert Feature Summary:.....	17
AOLP Flags:.....	19
Email templates:	211
JSON variable queries:	29
Possible State & their weightage:.....	31
InfoSync URL:.....	32
File ID conversion table:.....	33
Formdef – Change Specs:.....	34

Synopsis

Advance Online Proofing (AOLP) is an enhanced version of Online Proofing; with AOLP, entire job cycle (regardless of print or E) is available for proofing online for statements, notices or taxes. Upon processing, the financial institution is initially notified once the cycle is released for proofing, as well as prompted for any proofing approvals that are due afterwards. Using the InfoTRAC job tracking system, the financial institution counterparts can drill down to view individual document outputs within a cycle, and once satisfied with the review, can provide approval for the cycle.

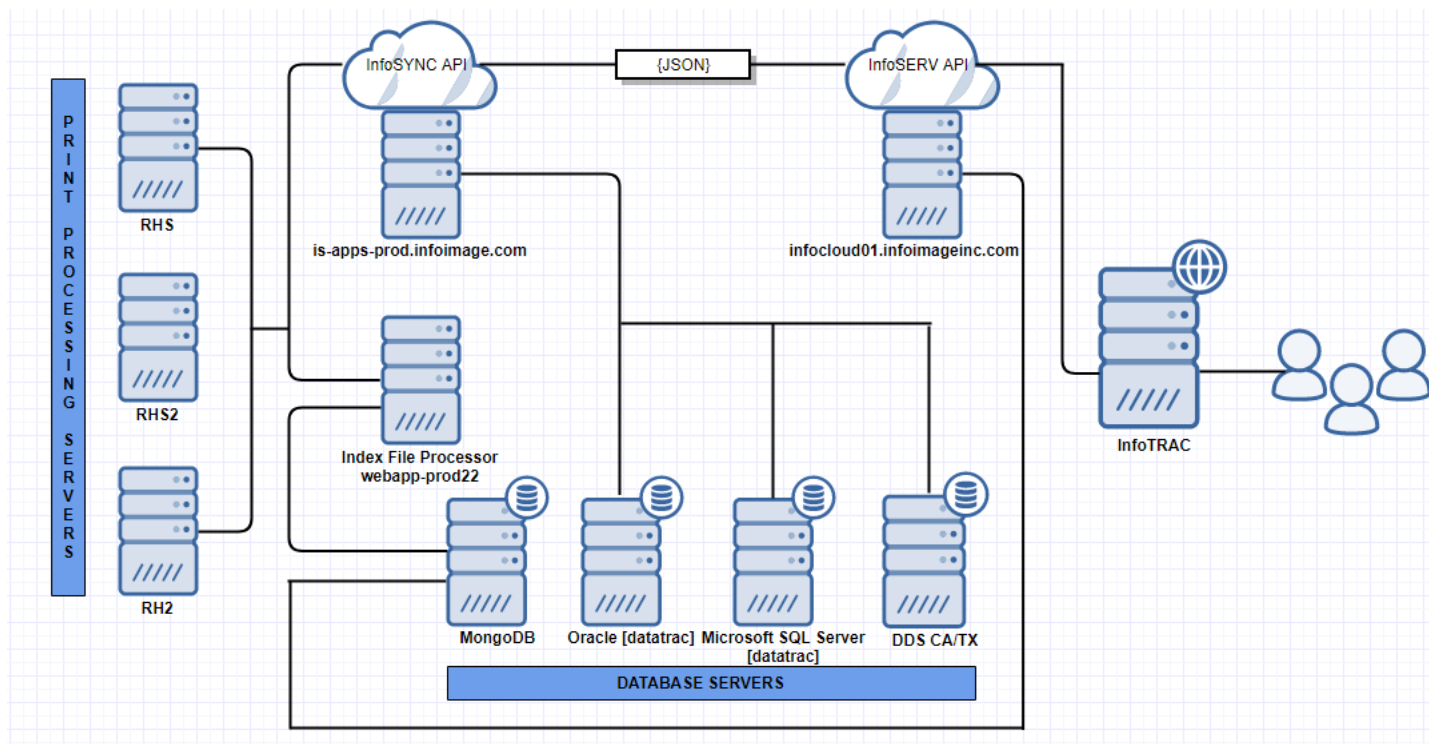
There is ability to search documents within a cycle by account or SSN if applicable. In addition, there is an option to have a targeted sample output file be available against a list of specific accounts for a print job for an added fee (quoted per hour). Our role-based access for the InfoTRAC platform can control which FI users can perform online proofing/approval.

Divert feature is automatically enabled with Advanced Online Proofing. This feature enables the financial institution to pull or divert individual mail pieces from our print stream as a self-service feature. Via the extended tracking for statements, notices or taxes. The financial institution counterparts can drill down within a particular cycle, select one or more documents and initiate the pull. InfoTRAC triggers the selected documents to be pulled from the print queue and automates the entire backend process of diverting the selected documents from our camera system, adjusting the checks/balances, updating the counts as well as billing data properly.

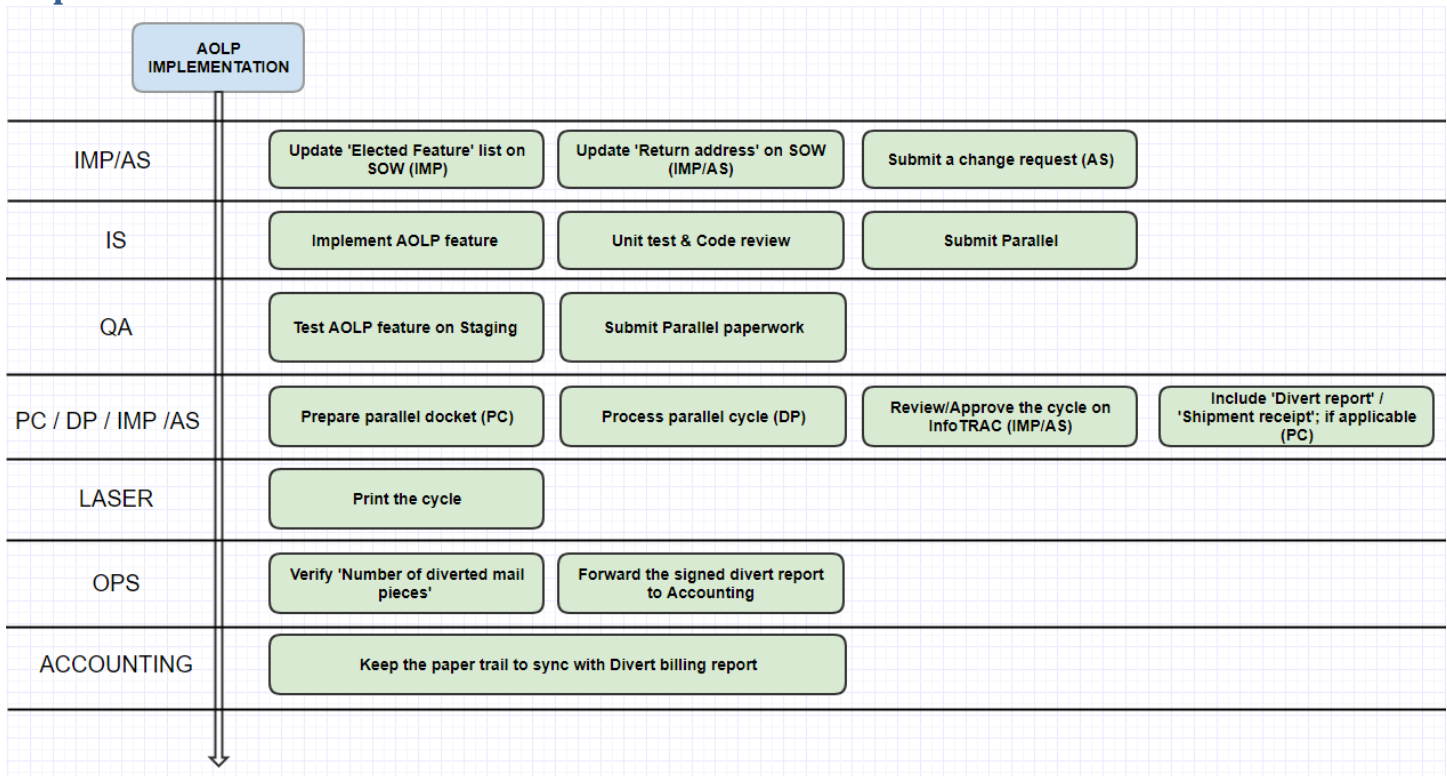
This guide provides detail specification for the application development on print side. For script maintenance details and future development plans, please consult with your manager.

Design Description

The diagram below details the essential outline of this design.

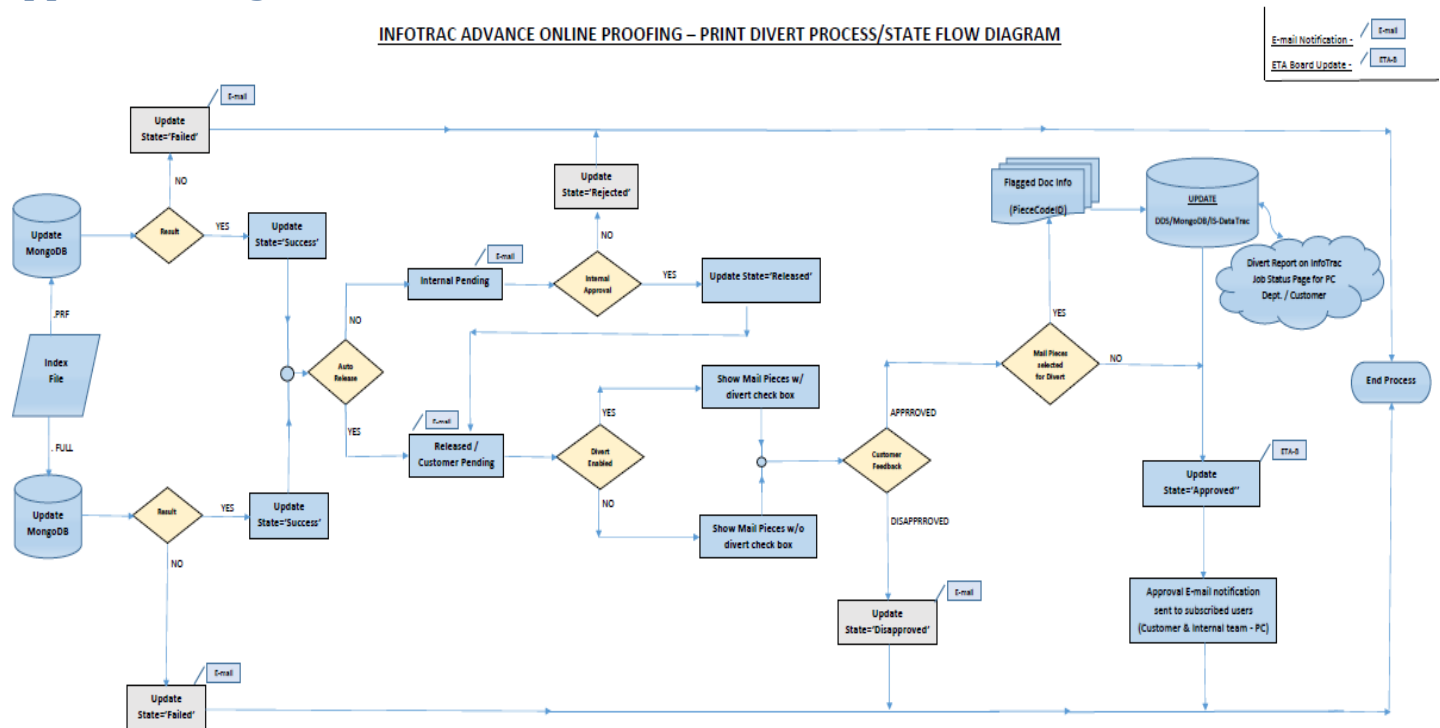


Implementation workflow



Application Logic - Flowchart

INFOTRAC ADVANCE ONLINE PROOFING – PRINT DIVERT PROCESS/STATE FLOW DIAGRAM



IS Database Schema changes

- Add columns to an existing **PROCESS_MAIN** table to reference **PROCESS_MAIN_OLP** table:

- **TABLE: PROCESS_MAIN**

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	osculb11720181217160906.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 generated when record is inserted

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

- 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	<PROC_ID>
TIMESTAMP	DATE	TIMESTAMP
SERVICE	VARCHAR2	onlineProofing/extendedMailTracking
PRF_STATE	VARCHAR2	
FULL_STATE	VARCHAR2	
MAIN_STATE	VARCHAR2	
WARNING_CODE	NUMBER	000
WARNING_DESC	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

- Create **OLP_DIVERT_SUMMARY** table to record all the pulls for reporting purpose:

○ **TABLE: OLP_DIVERT_SUMMARY**

COLUMNS	TYPE	VALUE	SOURCE
OLP_ID	NUMBER	Optional	Process_Main
JOB_2D	CHAR	Required	First 3 chars of mail_piece_id
CID	VARCHAR2	Required	Process_Main
MAIL_PIECE_ID	VARCHAR2	Required	StateUpdate API parameter
DT_JOB_ID	VARCHAR2	Required	Process_Main
USERID	VARCHAR2	Required	StateUpdate API URL parameter
PULL_STATUS	VARCHAR2	Required	Based on DDS update
PULL_INSTRUCTION	VARCHAR2	Required	StateUpdate API parameter
LAST_UPDATED	DATE	Required	Database timestamp
ADDITIONAL_INSTRUCTION	VARCHAR2	Optional	StateUpdate API parameter <TBD>
ACCOUNT_NUMBER	VARCHAR2	Optional	Extract from DP_Rawinfo_<job_2d> table
ACCOUNT_NAME	VARCHAR2	Optional	Extract from DP_Rawinfo_<job_2d> table
ADDRESS1	VARCHAR2	Optional	Extract from DP_Rawinfo_<job_2d> table
ADDRESS2	VARCHAR2	Optional	Extract from DP_Rawinfo_<job_2d> table
ADDRESS3	VARCHAR2	Optional	Extract from DP_Rawinfo_<job_2d> table
ADDRESS4	VARCHAR2	Optional	Extract from DP_Rawinfo_<job_2d> table
ADDRESS5	VARCHAR2	Optional	Extract from DP_Rawinfo_<job_2d> table

○ **TABLE: MM_PROCESS_COUNT**

COLUMNS	TYPE	VALUE	SOURCE
ID	NUMBER	Required	
PROC_ID	NUMBER	Required	API parameter
MSG_BOX_NUM	NUMBER	Required	API parameter
CRITERIA_TYPE	VARCHAR2	Required	API parameter
MSG_AREA_TYPE	VARCHAR2	Required	API parameter
COLOR	CHAR(1)		API parameter
MSG_COUNT	NUMBER	Required	API parameter
MESSAGE_ID	NUMBER	Required	API parameter
HEIGHT	NUMBER		
WIDTH	NUMBER		

○ **TABLE: MM_DIMENSION**

COLUMNS	TYPE	VALUE	SOURCE
ID	NUMBER		
MSG_FORMAT	VARCHAR2		Manual
MAX_AREA	NUMBER		Manual
ACTIVE	VARCHAR2		Manual
MODIFIED	VARCHAR2		Manual

Process Stages / E-mail templates

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

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)

Environment & Folder Hierarchy

- **Processing servers:**
 - is-apps-prod.infoimageinc.com (Production) - 10.8.9.45
 - is-apps-stage.infoimageinc.com (Staging) - 10.8.9.164
 - is-apps-dev.infoimageinc.com (Pilot) - 10.8.9.119

- **Code base: /var/www/InfoSync**
 - o InfoTrac
 - ExtendedMailTracking
 - OLP
 - Divert
 - MM
 - o EtaBoard
 - DashBoard
 - o DataTrac
- **Logs: /var/log/infosync/infosync.log**

API specs & operations

InfoSync/InfoTrac/OLP/StateUpdate API

Operations:

- **[POST (INSERT)];**
 - o 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)
 - o 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
 - o 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)]**
 - o 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)
 - o 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
 - o 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

<u>Parameters</u>	<u>Required (*)</u>	<u>Description</u>
proc_id	*	Unique id for processing job
cid	*	3-character id assigned to job
state	*	Current state (ref state table)
userId	*	User who submitted the request
state_desc		State description
olp		Unique AOLP id
olp_prf_file		Flag for proof file
olp_auto_rel		Flag for auto release
olp_int_dlist		Internal email distribution list
olp_divert		Flag for divert feature
message		Message
warning_code		Warning code when state is set to warning
warning_desc		Warning description
prf_txt_name		Proof file txt name
full_txt_name		Full file txt name

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.

<u>Parameters</u>	<u>Required (*)</u>	<u>Description</u>
proc_id	*	Unique id for processing job

cid	*	3-character id assigned to job
-----	---	--------------------------------

InfoSync/InfoTrac/ExtendedMailTracking/JobTracking API

Operations:

- 1.The JobTracking API will be developed under ExtendedMailTracking
- 2. If STATE value equals “submitted” OR “warning” OR “success” then Execute EMT_STATE logic

Parameters	Required (*)	Description
proc_id	*	Unique id for processing job
cid	*	3-character id assigned to job
state	*	Current state (ref state table)
userId	*	User who submitted the request
message		Message
warning_code		Warning code when state is set to warning
warning_desc		Warning description
prf_txt_name		Proof file txt name
full_txt_name		Full file txt name

InfoSync/InfoTrac/ExtendedMailTracking/GetJobState API

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:
 - o 2.1 If yes;
 - 2.1.1 Include the record in response JSON with OLP_ID value
 - o 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

Parameters	Required (*)	Description
state	*	Current state value (submitted, warning, success)
archivalPeriod		In days e.g. 30

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
 - o When status = 2 i.e. hold
 - auto generate the timestamp for time_hold column
 - update IFS_JOB_LIST record to set status = 2
 - o When status = 3 i.e. release
 - auto generate the timestamp for time_rel column
- 4. Rest endpoint call returns

Parameters	Required (*)	Description
proc_id	*	Unique id for processing job
status	*	Job status - hold(2)/release(3)
message		Message
rel_message		Warning code when state is set to warning
user_id_hold		Warning description
user_id_rel		Proof file txt name

- **Passing parameters based on state:**
 - o if main_State = "submit"
 - *proc_id: Unique Id
 - *status: 2
 - message: "Job submitted to IndexFileProcessor"
 - user_id_hold: \$user_id
 - o if main_State = "approved"
 - *proc_id: Unique Id
 - *status: 3
 - rel_message: "Job is approved by the customer"
 - user_id_rel: SysOper

InfoSync/InfoTrac/divert/GetDivertInfo API

Parameters	Required (*)	Description
dt_job_id	*	Unique job id for processing job
proc_id	*	Unique id for processing job

Body:

```
{
  "cid": "$cid",
  "appid": "$appid",
  "dt_job_id": "$dt_job_id",
  "proc_id": "$proc_id",
  "pull_instructions": "$pull_instructions",
  "total_divert_pieces": "$total_divert_pieces",
  "job_2d": "$job_2d",
  "acct_detail_values": [
    {
      "account_number", "$account_number",
      "account_name", "$account_name",
      "mail_piece_id", "$mail_piece_id"
    },
    {
      "account_number", "$account_number",
      "account_name", "$account_name",
      "mail_piece_id", "$mail_piece_id"
    }
  ]
}
```

All the values except below will be pulled from "olp_divert_summary" table:

- Appid – process_main_olp
- Proc_id – process_main
- Total_divert_pieces – process_main_olp

InfoSync/InfoTrac/divert/GetDivertReport API

Parameters	Required (*)	Description
cid	*	3-character cid assigned to job
Jid	*	3-character jid assigned to job
dateFrom		Beginning Date
dateTo		End Date

Note - default date range is 31 days

Body:

```
{
  "cid": "$cid",
  "appid": "$appid",
  "dt_job_id": "$dt_job_id",
  "proc_id": "$proc_id",
  "pull_instructions": "$pull_instructions",
  "total_divert_pieces": "$total_divert_pieces",
  "job_2d": "$job_2d",
  "acct_detail_values": [
    {
      "account_number", "$account_number",
      "account_name", "$account_name",
      "mail_piece_id", "$mail_piece_id"
    },
    {
      "account_number", "$account_number",
      "account_name", "$account_name",
      "mail_piece_id", "$mail_piece_id"
    }
  ]
}
```

Processing Steps

- On the IS side; prf files (Index txt, print txt & pdf) are copied to: /d/isd_share/<CID>/ folder in production; For parallel it will be under /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 (mandatory)
 - olp_prf_file = y/n (optional)
 - olp_auto_rel = y/n (optional)
 - olp_int_dlist = <grep 'olp_maillist ' from cidjid_mail.lis> (optional)
 - olp_divert = y/n (optional)
- For the **new jobs** the following fields/flags are 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'
 - prf_p*_ovl flags (if applicable) need to be present in order to pull pre-prints for proof file.
 - mail_full_serv must be set to 'ASE3' or 'ASE4'
 - AOLP feature would not work with DT_JOB_ID '**888888**' or '**999999**'
- Following steps are used 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)
 - 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

Pseudo code / EMT state logic:

- 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
 - o 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~~
 - o 1.2 If no;
 - 1.2.1 Insert a new record with given parameters
 - o 1.3 Create a row in PM_EVENT_LOG table with service value "extendedMailTracking"

Pseudo code / Message Manager Criteria:

- For each proc_id there can be multiple criteria
- For each criteria
 - o If msg_area_type is 'image'
 - Get height and width of the message_area with message_id from 'InfoTrac' Database
 - Calculate area with height and width and find out the corresponding msg_format type, from MM_DIMENSION table. This is calculated using are <= type
 - Example:

Image01	8
Image02	24
Image03	999999
 - Save result in MM_PROCESS_COUNT table
 - o Else Save result in MM_PROCESS_COUNT table

Divert feature summary

The current divert feature uses following two databases:

- DDS (For California print jobs)
- DDS_TX (For Texas print jobs)

During the paper processing for each jobId (5 digit ID) the following 6 tables are created in both the databases:

- dp_master_<JobId>
- dp_pockets_<JobId>
- dp_rawinfo_<JobId>
- dp_remedy_<JobId>
- dp_segment_<JobId>
- dp_jobid_<JobId>

For diverting the mail piece the following two tables are updated:

- dp_master_<JobId> (To update PullFlag column)
- dp_rawinfo_<JobId> (To update Pull_Description column)

InfoTRAC Paper divert:

- Based on job_2d from Index txt the 10 digit mailPieceCode will be loaded into MongoDB.
- UI will get the divert flag status from InfoSync GetState API & based on flag value divert check boxes will be shown against each mail piece. User will not see checkbox if the 10 digits code is missing in mongo for a document.
- The user may choose to select 1 or more mail piece for divert (Per Ops Max limit is set to 25) & would also provide pull description.
- The UI will pass the following JSON to InfoSync stateUpdate API with list of 10 digits mail piece code and the divert description (UI -> infoService -> InfoSync)

```
{  
    "state": "approved",  
    "message": "The job is approved",  
    "infotrac_login_id": "",  
    "comments": "",  
    "client_accountCoordinator_fullName": "",  
    "client_accountCoordinator_phone": "",  
    "client_accountCoordinator_email": "",  
    "error_message": "",  
    "divertPieces": [1234567891, 1234567892, 1234567893, 1234567894],  
    "divertDesc": "<UI description entered by user>"  
}
```

- Upon receiving the request; It will check if this job is already submitted with divert. If yes, then it resets the PullFlag to 0 and Pull_Description to "" of the corresponding dds database table. And then it also deletes those entries from the LP_DIVERT_SUMMARY from IS database.
- the 10 digit mailPieceCode will be parsed as jobId (First 5 chars) & seqNum (Last 5 chars).
- Based on jobId & seqNum the following update statements will execute on either CA & TX database:
 - o UPDATE dds.dbo.dp_master_<jobId> SET PullFlag = 1 WHERE SequenceID='<seqNum>';
- Run another update statement on following table:
 - o UPDATE dds.dbo.dp_rawinfo_<jobId> SET Pull_Description = '<diverDesc>' WHERE PieceID='<mailPieceCode>';
- Update Datatrac OLP_DIVERT_SUMMARY table
- Send 'Approval' email w/ divert info
- Update Divert information in MondoDB using following service:
 - o **Sample request:**
http://pd-pilot1.infoimageinc.com:8580/infoService/api/v1/onlineproofing/updateMongo?cid=demo&proc_id=639841

Body:

```
{
```

```

"values" : [
{
    "mail_piece_id": "ud14a00001 ",
    "pull_status"      : "success"
},
{
    "mail_piece_id": "ud14a00002 ",
    "pull_status"      : "unmark"
},
{
    "mail_piece_id": "ud15a00001",
    "pull_status"      : "success"
}
]
}

```

DDS Table model screenshot:

Columns						
ORDINAL_POSITION	COLUMN_NAME	DATA_TYPE	IS_NULLABLE	NUMERIC_PRECISION	NUMERIC_SCALE	DATETIME_PRECISION
1	AssignedID	int	NO	10		0 (null)
2	SequenceID	nvarchar	YES	(null)	(null)	(null)
3	PullFlag	int	YES	10		0 (null)
4	TimesSeen	int	YES	10		0 (null)
5	SeenFlag	int	YES	10		0 (null)
6	FirstSeenBy	int	YES	10		0 (null)
7	MatchFlag	int	YES	10		0 (null)

DDS_dbo.dp_master_s0nis	
P *	AssignedID INTEGER
	SequenceID NVARCHAR (32)
	PullFlag INTEGER
	TimesSeen INTEGER
	SeenFlag INTEGER
	FirstSeenBy INTEGER
	MatchFlag INTEGER
PK__dp_maste__4A3E3939234E0BD2 (AssignedID)	
SequenceID_Index (SequenceID)	
TimesSeen_Index (TimesSeen)	

dp_rawinfo_s0n4a

Start Page DDS datatrac DDS_TX dp_rawinfo_s0nis						
Columns Data Keys Indexes Constraints Triggers Model						
Actions...						
ORDINAL_POSITION	COLUMN_NAME	DATA_TYPE	IS_NULLABLE	NUMERIC_PRECISION	NUMERIC_SCALE	DATETIME_PRECISION
1	PieceID	nvarchar	YES	(null)	(null)	(null)
2	Account_Number	nvarchar	YES	(null)	(null)	(null)
3	Type	nvarchar	YES	(null)	(null)	(null)
4	Type_Description	nvarchar	YES	(null)	(null)	(null)
5	Name	nvarchar	YES	(null)	(null)	(null)
6	Address_1	nvarchar	YES	(null)	(null)	(null)
7	Address_2	nvarchar	YES	(null)	(null)	(null)
8	Address_3	nvarchar	YES	(null)	(null)	(null)
9	Address_4	nvarchar	YES	(null)	(null)	(null)
10	Address_5	nvarchar	YES	(null)	(null)	(null)
11	SequenceID	nvarchar	NO	(null)	(null)	(null)
12	Pull_Description	nvarchar	YES	(null)	(null)	(null)

DDS_dbo.dp_rawinfo_s0nis	
PieceID	NVARCHAR (16)
Account_Number	NVARCHAR (8)
Type	NVARCHAR (8)
Type_Description	NVARCHAR (40)
Name	NVARCHAR (40)
Address_1	NVARCHAR (24)
Address_2	NVARCHAR (24)
Address_3	NVARCHAR (8)
Address_4	NVARCHAR (8)
Address_5	NVARCHAR (8)
P * SequenceID	NVARCHAR (64)
Pull_Description	NVARCHAR (64)
PK__dp_rawin__BAD614F12AEFED9A (SequenceID)	

AOLP Flags

Parameter Name	Arguments	Description	Dependencies
Olp (mandatory)	y: yes n: no default: N/A	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
olp_prf_file (optional)	y: yes n: no default: n	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 (optional)	y: yes n: no default: y	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 (optional)	<email ids> default: DP email id	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 (optional)	y: yes n: no default: y	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

Email templates

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_accountCoordinator_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

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.

Ref#: \${dt_job_id}\${\$proc_id}

[JSON DATA]

```
"template_id": "jobStatusOLP.warning.notification",  
"email_list": [{ac@infoimageinc.com}, {dp@infoimageinc.com}],  
"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**[EMAIL TEMPLATE]****from:**Data.Center@infoimageinc.com**subject:**

Alert - INFOTRAC-Extended Job Status outstanding PieceCode issues

body:

Dear PC Department,

This is a reminder email with a list of jobs that are still pending "Piece Code" update:

PREFIX	APP_ID	DT_JOB_ID	PROC_ID	PENDING_FROM_DATE	REPROCESSING_LINK
\$cid\$jid	Letter - Daily	111111-C11.01-V11111	1111111	2000-01-01 00:00:01	Reprocess full file

After you've updated the Piece ID Code, please click on the reprocessing link above corresponding to the job; If you don't see the word "Success" in the browser, please immediately contact Product Development team to research this issue and include a screen shot of what you are seeing in the browser.

Thank You,

IS Team

InfoIMAGE Inc.

141 Jefferson Drive, Menlo Park, CA 94025 T. 650.473.6388 | F. 650.473.6300 | www.infoimageinc.com

Template ID: jobStatus.PendingJobs.notification

[EMAIL TEMPLATE]

from:

Data.Center@infoimageinc.com

subject:

Alert - INFOTRAC-Extended Tracking Jobs - Pending updates from IndexFileProcessor API

body:

Dear ISD Department,

The following jobs are in submit state for more than 60 minutes & have not received any response from IndexFileProcessor.

PREFIX	APP_ID	DT_JOB_ID	PROC_ID	PENDING_FROM_DATE	REPROCESSING_LINK
\$cid\$jid	Letter - Daily	111111-C11.01-V11111	1111111	2000-01-01 00:00:01	Reprocess full file

Please research & check the status

Thank You,

IS Team

InfoIMAGE Inc.

141 Jefferson Drive, Menlo Park, CA 94025 T. 650.473.6388 | F. 650.473.6300 | www.infoimageinc.com

JSON variable queries

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 tblAI
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>  
  
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');
```

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;
```

Possible State & their weightage

Below is the list of possible MAIN_STATE values:

Response
NULL
WARNING

SUBMITTED
FAILED
INTERNAL_PENDING
CUSTOMER_PENDING
RELEASED
REJECTED
APPROVED
DISAPPROVED
DELETE

Weight	State(s)
0	Null
1	Submitted
2	warning/success/internal_pending
3	released / customer_pending
4	Approved
5	failed/rejected/disapproved/delete

InfoSync 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

FormDef – Change Specs

Process/Tasks:

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

```
e_stmt_format_dfa="OSCUMS21"  
pdf_format_dfa="OSCUMS21"  
ind_pdf_format_dfa="OSCUMS21"  
print_dfa="OSCUMS22"  
print_formdef="FlXXXMS2.FDF"  
profile="/home/isis/userisis/pdec.prf"  
PP1="3061"  
PP2="4000"  
new_envelope_structure="y"  
ENV1="4401"  
ENV2="4402"
```

-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

```
service_id=041  
track_by_rec=y  
imb_track=y  
job_2d=rdz  
olp=y  
olp_divert=y  
FormDefPath=/home/test/isis/fdf_pdf/oscu/  
FORMDEFFILENAME=FlXXXMS2.FDF  
[IDX_FILE]  
oscums21154.2o1.20180726002508  
oscums2115e.sup.20180726002508  
oscums21155.npm.20180726002508
```

-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/oscu21155.2oz.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/  
oscu21155.2oz.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/oscu21155.2oz.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/  
oscu21155.2oz.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_parallel.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/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 (FRST ES3 example):

/home/isis/ppfa/frst/

/home/isis/fdf_pdf/frst/

Naming conventions (FRST ES3 example):

PPFA – FRST01.PFA

FORMDEF - F1FRST01.FDF