RQC Access APP

Service-Now

SharePoint

SN: Step 2

SN: Step 3

SN: Step 1

Timeline:

1. June 16 > Alpha testing
2. July 1rst > Beta testing
3. July 16 > Production release

TODO

1. SSIS Solutions Implementation
2. APP API Backend Setting
3. APP Implementation
4. APP Integration with SharePoint
   1. Test environment
   2. Production environment
5. APP Integration with Service-now
   1. Test environment
   2. Production environment

Next Steps:

1. SSIS
   1. Created a service account for SharePoint connections
      1. User: svc-finance\_etl-sp@jax.org
   2. Regarding Job Setup - how will these need to be run/scheduled? In any specific order, daily, weekly, etc.
      1. Job to run daily
   3. Set flat file connection managers to point to files stored under
      1. E:\SSIS\_Support\Finance (which will exist on all the servers this will run on)
   4. Configure the packages to be environment agnostic - Use variables in building SQL statements
      1. Added the following solution parameters:
         * INFORLAWSON\_LOGSCHEMA
         * INFORLAWSON\_LAWSCHEMA
         * INFORLAWSON\_PRODLINE
      2. To do:
         * SharePoint Connection manager – only One
         * Create solution wide parameters for SharePoint connection
           1. SharePoint\_UserName
           2. SharePoint\_Password
           3. SharePint\_ServerName
           4. SharePoint\_ModelsBase
         * Cleanup the solution connection manager
      3. Affected Database tables
         1. PRODLAW.GLNAMES -- accounting units
         2. LOGLAW.RMDATA -- employee info
         3. PRODLAW. ACACTIVITY
         4. PRODLAW.AUMXVALUE – employee title [ budget manager or FA)
         5. PRODLAW.RQLOC
         6. PRODLAW.RQACCTU -- accounting unit requester mapping
         7. PRODLAW.REQUESTER
      4. Affected models:

* Active Accounting Units SQL expression: Model not used

“select trim(description) as description, trim(acct\_unit) as acct\_unit, obj\_id, parent\_obj\_id

from "+@[$Project::INFORLAWSON\_LAWSCHEMA]+".GLNAMES where trim(active\_status) = 'A'

AND TRIM(posting\_flag) = 'P'

and trim(chart\_section) = 2

order by 1 asc”

* Active Employee SQL expression: (emp\_id, emp\_email, emp\_name, user\_name)

"SELECT distinct trim(employee) as emp\_id, trim(translate(ssop\_user,chr(10)||chr(11)||chr(13), ' ' )) as emp\_email,

translate(trim(CONCAT(CONCAT(TRIM(first\_name), ','), trim(last\_name))) ,chr(10)||chr(11)||chr(13), ' ' ) as emp\_name,

trim(translate(trim(rm\_id), chr(10)||chr(11)||chr(13), ' ' )) as user\_name

FROM "+ @[$Project::INFORLAWSON\_LOGSCHEMA] +".rmdata

where trim(product\_line)= )='" +@[$Project::INFORLAWSON\_PRODLINE] +"' and employee<>0"

* AccountingUnits2Activity mapping – SQL expression

select trim(ac.acct\_unit) as acct\_unit, trim(ac.activity) as activity,

trim(ac.description) as short\_desc , trim(ac.ACTIVITY\_GRP) as activity\_grp,

trim(ac.usr\_status) as usr\_status, trim(ac.BEGIN\_DATE) as begin\_date,

trim(ac.END\_DATE) as end\_date,

trim(CONCAT(CONCAT(ac.LONG\_DESC\_01,'-'), TRIM(ac.LONG\_DESC\_02))) as long\_desc

from "+@[$Project::INFORLAWSON\_LAWSCHEMA]+".ACACTIVITY ac, "+@[$Project::INFORLAWSON\_LAWSCHEMA] +".GLNAMES gl

where trim(ac.acct\_unit) = trim(gl.acct\_unit)

and ac.end\_date >= current\_date

and trim(gl.active\_status) = 'A'

AND TRIM(gl.posting\_flag) = 'P'

and trim(gl.chart\_section) = 2"

* AccountingUnits2BudgetManagerAndFinancialAnalyst – SQL Expression

"select trim(gl.ACCT\_UNIT) as acct\_unit, trim(a.MATRIX\_CAT) as matrix\_cat,

trim(ec.employee) as emp\_id, trim(ec. ssop\_user) as emp\_email,

trim(CONCAT(CONCAT(TRIM(ec.first\_name), ','), trim(ec.last\_name))) as emp\_name

from "+@[$Project::INFORLAWSON\_LAWSCHEMA]+".glnames gl inner join "+@[$Project::INFORLAWSON\_LAWSCHEMA]+".AUMXVALUE a on gl.OBJ\_ID = a.OBJ\_ID,"+ @[$Project::INFORLAWSON\_LOGSCHEMA] +".rmdata ec where a.MATRIX\_CAT

in ('FINALYST','BUDMGR')

and trim(gl.active\_status) = 'A'

AND TRIM(gl.posting\_flag) = 'P'

and trim(gl.chart\_section) = 2

and trim(SUBSTR(a.MX\_VALUE,1, INSTR(a.MX\_VALUE,'|', 1, 1)-1))= trim(ec.employee)

and ec.ssop\_user LIKE '%jax.org%'

and ec.employee <>0 order by 1"

* RQC Location SQL Expression

"select trim(REQ\_LOCATION) as req\_location, trim(R\_NAME) as r\_name, trim(CONCAT(CONCAT( TRIM(ADDR1), ', '), ADDR2)) as address

from "+ @[$Project::INFORLAWSON\_LAWSCHEMA] +".rqloc where trim(active\_status)='A' and TRIM(COMPANY)='1'"

* Active Requestors2Au SQL Expression

"select trim(rq.requester) as requester, trim(rq.acct\_unit) as acct\_unit,

trim(r.R\_NAME) as r\_name, trim(r.req\_location) as req\_location,

trim(r.require\_vendor) as require\_vendor,trim(r.email\_address) as email\_address

from "+ @[$Project::INFORLAWSON\_LAWSCHEMA] +".rqacctu rq,

"+ @[$Project::INFORLAWSON\_LAWSCHEMA] +".GLNAMES gl,

"+ @[$Project::INFORLAWSON\_LAWSCHEMA] +".REQUESTER r

where rq.company=1

and trim(rq.company)=trim(r.company)

and trim(rq.requester)=trim(r.requester)

AND TRIM(r.active\_status) = 'A'

and trim(rq.acct\_unit) = trim(gl.acct\_unit)

and trim(gl.active\_status) = 'A'

AND TRIM(gl.posting\_flag) = 'P'

and trim(gl.chart\_section) = 2"

* 1. Optimize json models – only fields used are include – this will reduce the memory footprint
     1. Active Employees – Done
     2. Active Accounting Units – Done
     3. Accounting Units to Activity – Done
     4. Accounting Units to Budget Managers/Financial Analysts – Done
     5. Requestor Default Location – Done
     6. Active Requestors -- Done
  2. Add a data conversion step to the json generation task – to remove line breaks within data model string
     1. Active Employees
* SQL task - SQL command: SELECT CAST((

SELECT DISTINCT emp\_username as user\_name,

emp\_email, emp\_name, emp\_id

FROM active\_employees

FOR JSON PATH ) as VARCHAR(MAX) ) as emp\_json

* Data conversion task and set the datatype to Unicode Text Stream (DT\_N…)
* File transfer task: configure mapping to map the converted output field
  + 1. Active Accounting Units
* SQL task - SELECT CAST((

SELECT DISTINCT acct\_unit, desccription as description FROM active\_au

FOR JSON PATH ) as VARCHAR(MAX) ) as au\_json

* Data conversion task and set the datatype to Unicode Text Stream (DT\_N…)
* File transfer task map
  + 1. Accounting Units to Activity
* SQL task - SELECT CAST((

SELECT DISTINCT acct\_unit, activity, short\_desc FROM au2activity

FOR JSON PATH ) as VARCHAR(MAX) ) as au2activity\_json

* Data conversion task and set the datatype to Unicode Text Stream (DT\_N…)
* File transfer task map
  + 1. Accounting Units to Budget Manager
* SQL Task -- SELECT CAST((

SELECT DISTINCT emp\_email, emp\_id , acct\_unit

FROM au2bmgrfa

FOR JSON PATH ) as VARCHAR(MAX) ) as au2bmgrfa\_json

* Data conversion task and set the datatype to Unicode Text Stream (DT\_N…)
* File transfer task map to the converted output field
  + 1. Requestor Default Location
* SQL Task -- SELECT CAST((

SELECT DISTINCT req\_location, r\_name, address

FROM req\_location

FOR JSON PATH ) as VARCHAR(MAX) ) as req\_location\_json

* Data conversion task and set the datatype to Unicode Text Stream (DT\_N…)
* File transfer task map to the converted output field
  + 1. Active Requestors
* SQL task -- SELECT CAST((

SELECT DISTINCT emp\_username as user\_name,

emp\_email, emp\_name, emp\_id

FROM active\_employees

FOR JSON PATH ) as VARCHAR(MAX) ) as emp\_json

* Data conversion task and set the datatype to Unicode Text Stream (DT\_N…)
* File transfer task map to the converted output field

1. APP API
   1. Json models – reformat the models to return json object from axios call rather than a string – which result in associated select lists being empty
2. APP Implementation
   1. Bundle form content into a preview component
   2. Create email template for service-now ticket submission
   3. Implement the form submit behavior
   4. Code refactoring and form validation
3. APP Integration with SharePoint
   1. Test environment : jaxspdev
      1. Install app to app catalog - <https://jaxspdev.sharepoint.com/sites/apps/AppCatalog/Forms/AllItems.aspx>
      2. Create a SharePoint page to host the app - <https://jaxspdev.sharepoint.com/sites/jaxspdev/SitePages/rqcapp.aspx>
   2. Production environment: <https://jacksonlaboratory.sharepoint.com/>
      1. Install app to app catalog
      2. Approve app permissions (User.Read, Sites.Read.All) using portal.azure.com
      3. Create a SharePoint page to host the app
      4. See Service-now tickets:
         * RITM0057113: Please give me the right permissions on the Financial Services Site to assist this group with their software services needs.
         * RITM0056281: Installing the App to prod
4. APP Integration with Service-now - Suzanne Prince and Patrick Chase
   1. **Service-now ticket:**
      1. RITM0049734: Integration with new RQC Access SharePoint Application.
   2. **Phase one**
      1. Lucie to schedule time with Suzanne for next week to test email into ServiceNow QA to see if that will work as an interim solution
         * Let Suzanne know of the next ticket be created after Lawson step is complete
      2. Patrick to create an Enhancement request to begin the story development process and get this work into the demand pipeline
      3. Suzanne to reach out to Acorio as well as research potential solutions using either API or Sharepoint plugin
   3. **Phase two:**

* We reviewed what happens in QA when your new Sharepoint request is received via email with no customization and outlined the instructions that would need to be given to the Service Desk so they process these requests correctly.  The result is not ideal, so I recommended that we start to build a simple Request Item for RQC Access that would be used to route the ticket to your group more easily.
  + Secenario 1 – NO dedicated RQC Access Request
    - NEW CALL received by Service Desk via inbound email action and Service Desk would need to follow the below work instructions:
    - Step 1: Change the "Caller" to match Requested By:  from the description
    - Step 2: Select "Request" and "Ad Hoc"
    - Step 3: Copy "subject" line shown in the description into the “short description” field and submit
    - Step 4: On the RITM, navigate down to the SCTASK and change the assignment group from Service Desk to IT-GBL-ADMINISTRATIVE APPLICATIONS
  + Scenario 2 – Dedicated Request Item for RQC Access Request in place with workflow
    - NEW CALL received by Service Desk via inbound email action and Service Desk would need to follow the below work instructions:
    - Step 1: Change the "Caller" to match Requested By:  from the description
    - Step 2: Select "Request" and "RQC Access Request"
    - Step 3: Copy "subject" line shown in the description into the “short description” field and submit
      * SCTASK would automatically route to IT-GBL-ADMINISTRATIVE APPLICATIONS
      * IT-GBL-ADMINISTRATIVE APPLICATIONS fulfills the task and closes it
      * This closes the RITM, and the final step of the workflow would send an email to the Purchasing team via email distribution list with the details needed for the next part of the process

NEXT STEPS:

* Lucie to open up testing in the Sharepoint system – it can send to jaxqa but nothing will be tested on the ServiceNow side for the next 2 weeks
* Use the 2 week time period to gather requirements for the RQC Access Request and pass to development
* Time to implement fully with ServiceNow will be approximately 3 – 4 weeks depending on our development capacity
* Suzanne to begin the Requirements workbook and set up review with Lucie for next week

Other discovery:

* During our testing, we discovered a discrepancy in Groups.  The Lawson team processes tickets assigned to IT-GBL-ADMINISTRATIVE APPLICATIONS.
* Currently, there are several workflows that send tickets to the Lawson team via IT-GBL-L2-LAWSON, which is an inactive group with no members.  This is the group that was indicated for go live as the Lawson group.
  + Suzanne to enter an Incident ticket to identify any workflows using IT-GBL-L2-LAWSON and change them to IT-GBL-ADMINISTRATIVE APPLICATIONS
  + Suzanne to reassign any existing tasks or tickets for IT-GBL-L2-LAWSON and change them to IT-GBL-ADMINISTRATIVE APPLICATIONS