



# Leveraging Automic/UC4 Automation at San Jacinto College

**Presented by:**

**Mark Deaver - Business Analyst**



# Background

San Jacinto College is transforming the way we do business by leveraging Automic/UC4 and Banner applications to automate current manual processes.

San Jacinto has been using Automic UC4 for the past 10 years.

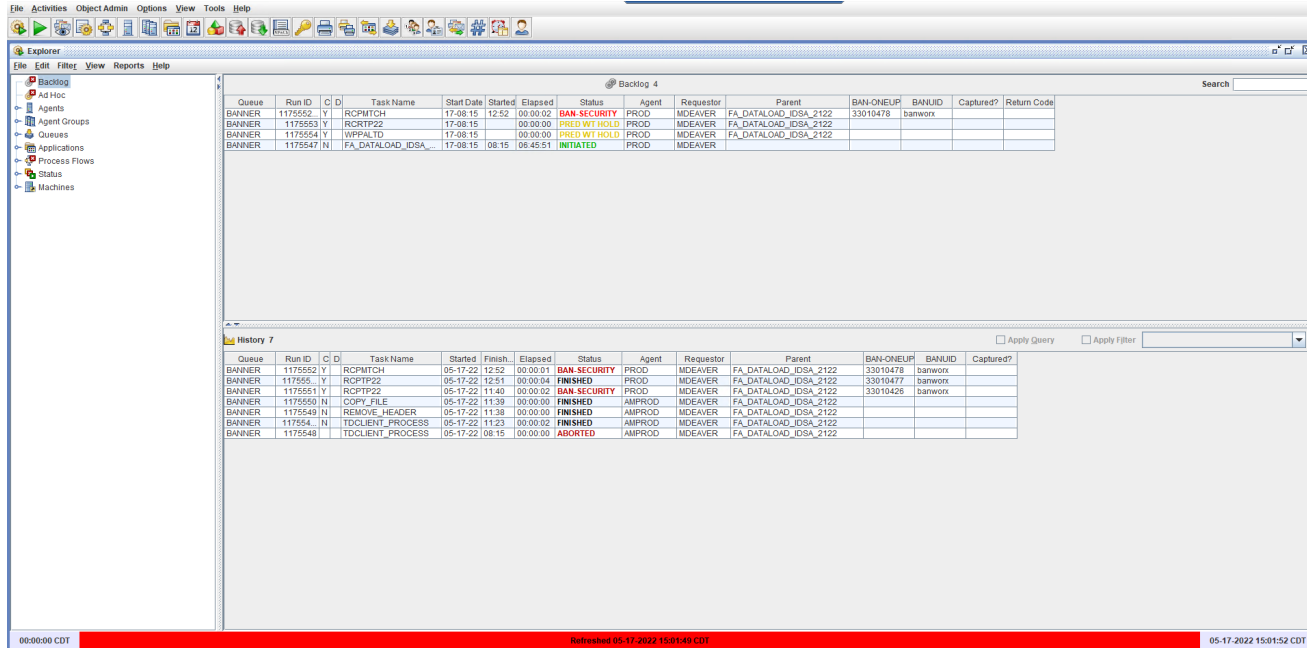
Discuss some of the key processing abilities in Automic Application Manager.

Demonstrate a few processes automated through Automic at San Jacinto College.

# Key Points

- Demonstrate the layout of the Automatic User Interface.
- Demonstrate how baseline jobs are processed through Automatic.
- Demonstrate how custom processes are executed through Automatic.
- Demonstrate how custom scripts can be integrated into Automatic.
- Demonstrate some of the processes San Jacinto automated in Automatic.

# Automic User Interface



The screenshot displays the Automic User Interface (UI) for a system named 'Backlog 4'. The interface includes a menu bar (File, Activities, Object Admin, Options, View, Tools, Help), a toolbar with various icons, and a left-hand navigation pane with categories like Backlog, Ad Hoc, Agents, Agent Groups, Queues, Applications, Process Flows, Status, and Machines.

The main area shows two tables: 'Backlog 4' and 'History 7'. Both tables have columns for Queue, Run ID, C, D, Task Name, Start Date, Started, Elapsed, Status, Agent, Requestor, Parent, BAN-ONEUP, BANUID, Captured?, and Return Code.

**Backlog 4 Table Data:**

Queue	Run ID	C	D	Task Name	Start Date	Started	Elapsed	Status	Agent	Requestor	Parent	BAN-ONEUP	BANUID	Captured?	Return Code
BANNER	1175552	Y		RCMPATCH	17-08-15	12:52	00:00:02	BAN_SECURITY	PROD	MDEAVER	FA_DATALOAD_IDSA_2122	33010478	banwon		
BANNER	1175553	Y		RCPTP22	17-08-15		00:00:00	PRED WY HOLD	PROD	MDEAVER	FA_DATALOAD_IDSA_2122				
BANNER	1175554	Y		WPPALTD	17-08-15		00:00:00	PRED WY HOLD	PROD	MDEAVER	FA_DATALOAD_IDSA_2122				
BANNER	1175547	N		FA_DATALOAD_IDSA_...	17-08-15	08:15	08:45:51	INITIATED	PROD	MDEAVER					

**History 7 Table Data:**

Queue	Run ID	C	D	Task Name	Started	Finish	Elapsed	Status	Agent	Requestor	Parent	BAN-ONEUP	BANUID	Captured?
BANNER	1175552	Y		RCMPATCH	05-17-22	12:52	00:00:01	BAN_SECURITY	PROD	MDEAVER	FA_DATALOAD_IDSA_2122	33010478	banwon	
BANNER	117555	Y		RCPTP22	05-17-22	12:51	00:00:04	FINISHED	PROD	MDEAVER	FA_DATALOAD_IDSA_2122	33010477	banwon	
BANNER	1175551	Y		RCPTP22	05-17-22	11:40	00:00:02	BAN_SECURITY	PROD	MDEAVER	FA_DATALOAD_IDSA_2122	33010426	banwon	
BANNER	1175550	N		COPY_FILE	05-17-22	11:39	00:00:00	FINISHED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			
BANNER	1175549	N		REMOVE_HEADER	05-17-22	11:38	00:00:00	FINISHED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			
BANNER	117554	N		TDCLIENT_PROCESS	05-17-22	11:23	00:00:02	FINISHED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			
BANNER	1175548			TDCLIENT_PROCESS	05-17-22	08:15	00:00:00	ABORTED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			

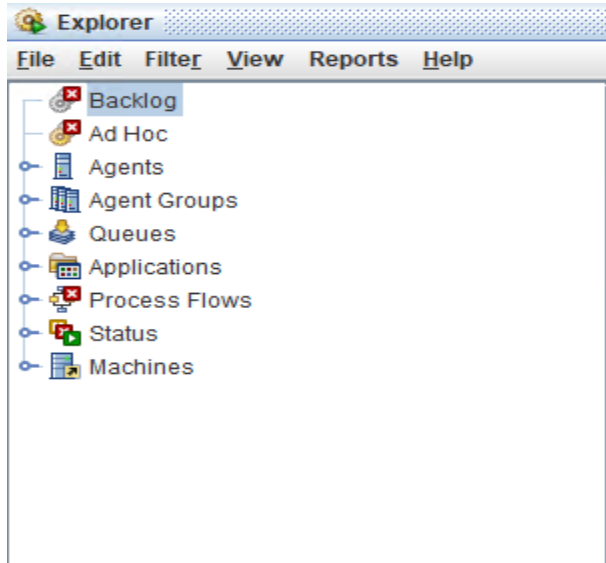
The interface also features a status bar at the bottom with the text '06:00:00 CDT', 'Refreshed 05-17-2022 15:01:48 CDT', and '05-17-2022 15:01:52 CDT'.

# Automatic User Interface – Menu/Toolbar



- Menu drop down options.
  - Administration options
  - Forecasting options
  - Other options
- Toolbar – Shortcut to commonly used options
  - Jobs & Process
  - Users
  - Substitution Variables

# Automatic User Interface – Explorer Window



- Explorer Window
  - Used to view items in:
    - Backlog
    - Ad Hoc
    - Agent Groups
    - Queues
    - Etc.
  - Select option to filter items in the Backlog or History window.
  - Selection of various reports available.
  - Look up general help.

# Automic User Interface – Backlog View

Backlog 4

Search

Queue	Run ID	C	D	Task Name	Start Date	Started	Elapsed	Status	Agent	Requestor	Parent	BAN-ONEUP	BANUID	Captured?	Return Code
BANNER	1175552...	Y		RCPMTC	17-08-15	12:52	00:00:02	BAN-SECURITY	PROD	MDEAVER	FA_DATALOAD_IDS_2122	33010478	banworx		
BANNER	1175553	Y		RCRTP22	17-08-15		00:00:00	PRED WT HOLD	PROD	MDEAVER	FA_DATALOAD_IDS_2122				
BANNER	1175554	Y		WPPALTD	17-08-15		00:00:00	PRED WT HOLD	PROD	MDEAVER	FA_DATALOAD_IDS_2122				
BANNER	1175547	N		FA_DATALOAD_IDS_...	17-08-15	08:15	07:00:02	INITIATED	PROD	MDEAVER					

- Backlog Window
  - Displays the jobs currently running.
  - Displays all jobs set to be executed.
  - Displays the current status of the jobs.
    - ABORTED
    - BAN-SECURITY
    - HOLD
    - INITIATED

# Automic User Interface – History View

History 7 ☐ Apply Query ☐ Apply Filter

Queue	Run ID	C	D	Task Name	Started	Finish...	Elapsed	Status	Agent	Requestor	Parent	BAN-ONEUP	BANUID	Captured?
BANNER	1175552	Y		RCPMTCH	05-17-22	12:52	00:00:01	BAN-SECURITY	PROD	MDEAVER	FA_DATALOAD_IDSA_2122	33010478	banworx	
BANNER	117555...	Y		RCPPT22	05-17-22	12:51	00:00:04	FINISHED	PROD	MDEAVER	FA_DATALOAD_IDSA_2122	33010477	banworx	
BANNER	1175551	Y		RCPPT22	05-17-22	11:40	00:00:02	BAN-SECURITY	PROD	MDEAVER	FA_DATALOAD_IDSA_2122	33010426	banworx	
BANNER	1175550	N		COPY_FILE	05-17-22	11:39	00:00:00	FINISHED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			
BANNER	1175549	N		REMOVE_HEADER	05-17-22	11:38	00:00:00	FINISHED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			
BANNER	117554...	N		TDCLIENT_PROCESS	05-17-22	11:23	00:00:02	FINISHED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			
BANNER	1175548			TDCLIENT_PROCESS	05-17-22	08:15	00:00:00	ABORTED	AMPROD	MDEAVER	FA_DATALOAD_IDSA_2122			

- History Window
  - Displays all finished jobs.
  - Also displays jobs that completed with other than a FINISHED status.
    - ABORTED
    - BAN-SECURITY



# Baseline job processing in Automic.

- Before a Banner process can be executed in Automic, the Banner job must be added in Automic.
- Banner Jobs are added to Automic through the Agent Administration in the Banner Tab under the Create Job tab.
- Once the desire job is created, click Cancel to exit out of the Agent Administration window.
- Note: The job object will need to be added to the Banner Security class of the UC4 user or the process will abort with a BAN-SECURITY error message.

# Baseline job processing in Automic.

Agents: PROD

General Banner User Groups

Agent Runtime Values Job Capture Create Job

Template Job BANNER\_TEMPLATE

Program	Description	Type	Command Na...	Existing
zfpjdt	Journal Vouch...	Procedure		
aapackn	Membership ...	Pro*C		
aapadjs	Dues Adjustm...	Pro*C		
aapcard	Membership ...	Pro*C		
aapfed	Advancement	Pro*C		
aapremd	Membership ...	Pro*C		
aapnew	Membership ...	Pro*C		
aapstat	Membership ...	Pro*C		
adpacct	Designation A...	Pro*C		
adpcfae	CFAE Report	Pro*C		
adpexpd	Expired Desig...	Pro*C		
adpfed	Advancement	Pro*C		
adppfed	Pledge Feed t...	Pro*C		
adpvser	Voluntary Sup...	Pro*C		
afpcamr	Campaign Hi...	Pro*C		
afpdonr	Campaign Do...	Pro*C		
afpsola	Campaign So...	Pro*C		
afpsob	Campaign Un...	Pro*C		
afpsoc	Campaign Un...	Pro*C		
afptelf	Campaign Tel...	Pro*C		

Search  ☐ Exclude Existing Jobs

New Job Name (PROGRAM)

Create Jobs

OK Apply Cancel Help

# Baseline job processing in Automic.

- Banner jobs will function in a similar manner as they would in Banner Job Submission.
- Jobs can use the exact same parameters they would be used in Banner.
- All job output is viewable in Automic.
- Job output can be emailed to a user.
  - Output can be sent in original format.
  - Output can be sent formatted in PDF.
  - Output can be sent in zip files.

# GLRLETR Example Job

Task Details: GLRLETR-FA\_APP\_REC'D\_ACK-2122 1175663

General Banner Predecessors Successors Flow Diagram Conditions Documentation Output Files Comments Comments Query

**Program Information**

Program:  \*

Command Name:

Program Type:  \*

Type Code:  \*

Login Override:

Source File Override:

**Banner Output**

Printer:

Printer Form:

Lines Per Inch:

**Advanced Output**

Output Search Pattern:

Rename Output Pattern:

**Parameters**

Prompt	Type	Description	Required	Value
01	Single	Application Code	Y	FINAID
02	Single	Word Processor Extract Option	Y	1
03	Single	Print ALL Pending Letters	N	
04	Single	Letter Code		FA_APP_REC'D_ACK
05	Single	Sort Variable		
06	Single	Term Code	Y	#2
07	Single	Module Code	Y	R
08	Single	Audit Indicator		N
09	Single	Free Format Date 1		{#sysdate}
10	Single	Free Format Date 2		{#sysdate}
11	Single	Free Format Date 3		{#sysdate}
12	Single	Aid Year Code		#1

**Parameter Set**

User Name:

Parameter Set:

☒ Allow Change at Request Time

☐ Preserve Existing Parameter Values

# GLRLETR Example Job

- Parameters can be made up of:
  - Hard coded values
  - Substitution variables
  - Prompt variables
  - Imported values from a saved data set
  - Dynamic Parameters
- Banner output can be set up to be sent to a printer.
- FormFusion Print Parameters can be used for the call to Evisions.



# GLRLETR Example Job

- Conditions can be added that will send job status notifications.
- Job output can be included in notification.
- Conditions can also be added to control a variety of tasks.
  - Check for file status.
  - Check for population counts.
  - Set substitution variables.
  - Run Host commands.



# GLRLETR Example Job

Task Details: GLRLETR-FA\_APP\_REC'D\_ACK-2122 1175663

General	Banner	Predecessors	Successors	Flow Diagram	Conditions	Documentation	Output Files	Comments	Comments Query		
Order				Timing	Performed		Condition#1	Qual	Condition#2	Action	Action Argument
1	BEFORE				ONCE		#FA_GLBDATA_top_flow_id	=	0	SEND NOTIFICATION	FA_TRACKING_LETTERS A
2	BEFORE				ONCE		#FA_GLBDATA_top_flow_id	=	0	SET SKIP STATUS	SKIP=0
3	AFTER				ONCE		STATUS	=	FINISHED	SEND NOTIFICATION	FA_TRACKING_LETTERS C

Condition Details - 1175663

Timing: AFTER Condition: STATUS

Test:

If status = FINISHED

Action:

Command: SEND NOTIFICATION FA\_TRACKING\_LETTERS C

Action Timing:

☒ The first time the condition is true

☐ Every time the condition is true

☐ Disabled

OK Cancel

New Edit Delete Copy... Up Down

OK Apply Cancel Help

# Baseline job processing in Automic.

- Examples of ideal processes in Automic:
  - Letter Generation
  - Term Grade Roll
  - Financial Aid Dataload
  - Financial Aid Processing
  - Automated Hold Processing



# Baseline job processing in Automic.

- Almost any Banner job can be automated in Automic.
- A rule of thumb would be to automate jobs that do not require frequent changes.
- Automating jobs that have static variables or static values are a great candidate for automation.
- For parameters or values that change, variables can be created and populated to handle some type of values.
  - Substitution variables can store the values.
  - Prompt Variables can store values for the specific job.
  - Conditions can be set up perform basic changes.

# Custom job processing in Automic.

- Custom jobs can be executed in Automic.
- Custom job should be coded in a similar fashion as a regular Banner job.
- Custom job would need to be submittable through Banner Job Submission.
- Output can be included in notifications as well.
  - Other functions may need to be programmed if the output is stored is not stored in the banjob directory.

# Custom job processing in Automic.

- Custom processes at San Jacinto typically start with ‘Z’.
- Output from job goes to a specific directory outside of normal banjob directory.
- An additional job is used to copy output file to the banjob directory.
- The copy file process uses an internal mail-x command to email the output files to users.

# ZFRZERO Custom job process

**ZFRZERO**

General Output **Banner** Predecessors Conditions Documentation

**Program Information**

Program  \*

Command Name

Program Type  \*

Type Code  \*

Login Override

Source File Override

**Banner Output**

Printer

Printer Form

Lines Per Inch

**Advanced Output**

Output Search Pattern

Rename Output Pattern

**Parameters**

Prompt	Type	Description	Required	Value
10	Single	Term code	Y	%

**Parameter Set**

User Name

Parameter Set

☒ Allow C

☐ Preset

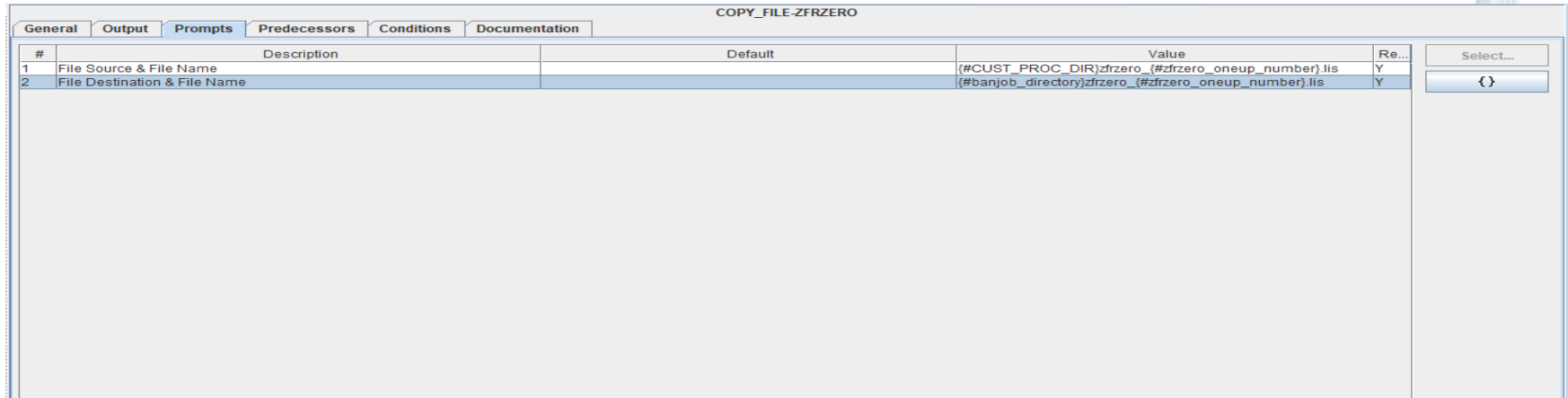
# ZFRZERO Example Job

- Conditions can be added that will send job status notifications.
- Conditions can also be added to control a variety of tasks.
  - Check for file status.
  - Check for population counts.
  - Set substitution variables.
  - Run Host commands.

# ZFRZERO - Conditions

ZFRZERO							
General	Output	Banner	Predecessors	Conditions	Documentation		
Order	Timing	Performed	Condition#1	Qual	Condition#2	Action	Action Argument
1	AFTER	ONCE	{status}	=	FINISHED	SEND NOTIFICATION	AR_ZFRZERO_NOTIFY F
2	AFTER	ONCE	{status}	!=	FINISHED	SEND NOTIFICATION	AR_ZFRZERO_NOTIFY A
3	AFTER	ONCE	STATUS	=	FINISHED	SET SUBVAR	#zfrzero_oneup_number={...

# ZFRZERO Copy File process



The image shows a screenshot of a software window titled "COPY\_FILE-ZFRZERO". The window has a tabbed interface with the following tabs: General, Output, Prompts, Predecessors, Conditions, and Documentation. The "Prompts" tab is currently selected. Below the tabs is a table with the following columns: #, Description, Default, Value, and Re... (likely Reusable). The table contains two rows of data. To the right of the table is a "Select..." button and a button with curly braces "{}".

#	Description	Default	Value	Re...
1	File Source & File Name		{#CUST_PROC_DIR}zfrzero_{#zfrzero_oneup_number}.lis	Y
2	File Destination & File Name		{#banjob_directory}zfrzero_{#zfrzero_oneup_number}.lis	Y

# ZFRZERO Copy File Process

- Copy File process generates a shell script to copy the output from custom process directory to the Banner banjob directory.
- Conditions are added to sent the notification of job status with the output.
- Automic is set up to only read the normal banjob directory.



# ZFRZERO – Copy File Conditions

COPY\_FILE-ZFRZERO

Order	Timing	Performed	Condition#1	Qual	Condition#2	Action	Action Argument
1	AFTER	ONCE	STATUS	=	FINISHED	RUN HOST COMMAND	echo "The output for the UC..."

**Condition Details - AR\_SEQUENCE\_OF\_REFUNDS(COPY\_FILE-ZFRZERO)**

Timing: AFTER  
Condition: STATUS

Test

If status = FINISHED

Action

Command: RUN HOST COMMAND  
Then: echo "The output for the UC4 automated"

Action Timing

- ☒ The first time the condition is true
- ☐ Every time the condition is true
- ☐ Disabled

OK Cancel

# Custom script processing in Automic.

- Custom scripts can be processed in Automic.
- Custom scripts can include:
  - Remove header records
  - Dos to Unix file format
  - File transfer processes
  - Insert header records
  - Parsing out a file into multiple files



# Custom script processing in Automic.

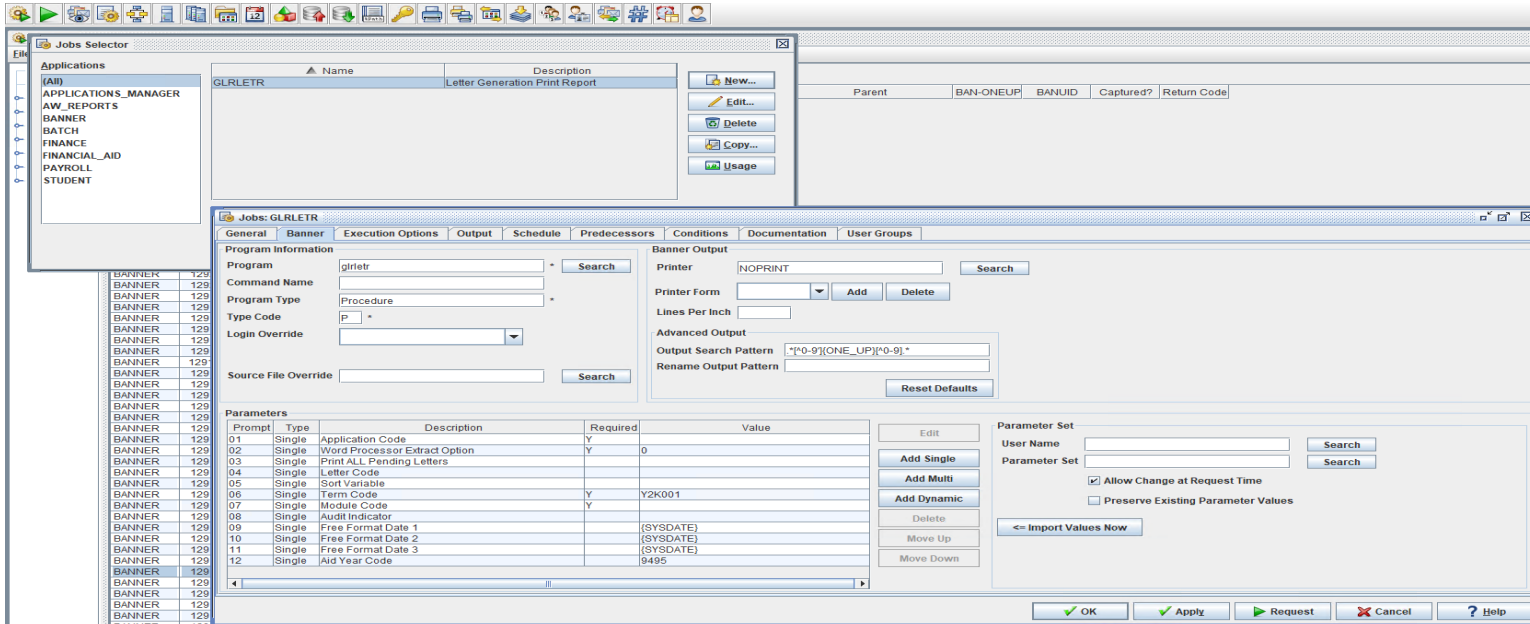
- Various types of scripts can be used.
  - Perl scripts
  - Shell scripts
  - etc
- Scripts are stored in the Automic exec directory.
- An Automic Job is created to process the script.
- The executable script is then assigned to the Automic job.



# Usage of Jobs in Automic/UC4

- Baseline/Custom Banner Jobs will exist as a stand-alone global job instance.
- The global job can be executed in a single run fashion using specified parameters.
- Be careful when using a global job as any modification to the global job instance could impact another process referencing the global job.
- It is advisable to add the global job to a Process.
- Once the job is in a Process, parameters, condition, etc can be added to job without affecting the global job.
- The parameters, conditions, etc assigned in a Process will only exist in the Process and not affect the global job.

# Global Job Instance



The screenshot shows the Banner Jobs Selector and the Jobs: GLRLETR form. The Jobs Selector on the left lists applications under the (All) category, including APPLICATIONS\_MANAGER, AW\_REPORTS, BANNER, BATCH, FINANCE, FINANCIAL\_AID, PAYROLL, and STUDENT. The main form displays details for the GLRLETR job, which is a Letter Generation Print Report. The form includes tabs for General, Banner, Execution Options, Output, Schedule, Predecessors, Conditions, Documentation, and User Groups. The General tab is active, showing program information, command name, program type, type code, login override, and source file override. The Banner tab shows printer, printer form, lines per inch, and advanced output settings. The Parameters tab shows a list of parameters with their prompts, types, descriptions, required status, and values. The Parameter Set tab shows user name, parameter set, and options to allow change at request time or preserve existing parameter values. The form also includes buttons for New, Edit, Delete, Copy, Usage, Add Single, Add Multi, Add Dynamic, Delete, Move Up, Move Down, OK, Apply, Request, Cancel, and Help.

**Jobs Selector**

Applications

(All)

APPLICATIONS\_MANAGER

AW\_REPORTS

BANNER

BATCH

FINANCE

FINANCIAL\_AID

PAYROLL

STUDENT

**Jobs: GLRLETR**

General Banner Execution Options Output Schedule Predecessors Conditions Documentation User Groups

Program Information

Program glrltr \* Search

Command Name

Program Type Procedure \* Search

Type Code P \* Search

Login Override

Source File Override Search

Banner Output

Printer NOPRINT Search

Printer Form Add Delete

Lines Per Inch

Advanced Output

Output Search Pattern \*C\*0-9(ONE\_UP)\*C\*0-9\* Search

Rename Output Pattern

Reset Defaults

Parameters

Prompt	Type	Description	Required	Value
01	Single	Application Code	Y	
02	Single	Word Processor Extract Option	Y	0
03	Single	Print All Pending Letters		
04	Single	Letter Code		
05	Single	Sort Variable		
06	Single	Term Code	Y	Y2K001
07	Single	Module Code	Y	
08	Single	Audit Indicator		
09	Single	Free Format Date 1		(SYSDATE)
10	Single	Free Format Date 2		(SYSDATE)
11	Single	Free Format Date 3		(SYSDATE)
12	Single	Aid Year Code		9495

Parameter Set

User Name Search

Parameter Set Search

☒ Allow Change at Request Time

☐ Preserve Existing Parameter Values

<= Import Values Now

OK Apply Request Cancel Help

# Process Instance

- A Process is like a string or chain and are made up of multiple jobs, scripts, or other processing functions.
- Multiple similar jobs can exist in one Process using different parameter sets.
  - Once a job is added to the Process, the parameters, condition, etc. are added to the job in the process.
  - These parameters, conditions, etc will only affect that specific job in the Process.
- Parameters saved for the job in a Process will not affect the global job instance.
- Processes offer a way to run multiple items to all a wide array of automation.

# San Jacinto Automatic Process Examples

- Part of Term Grade Roll
- Refunding processing for Finance Department
- Financial Aid Dataload processing



# Part of Term Grade Roll

Process Flows: ST\_PART\_TERM\_GRADEROLL

General Components Output Schedule Prompts Conditions Documentation User Groups

File Edit View Options Predecessors Simulation

Scale: 4

ST\_PART\_TERM\_GRADEROLL

External References

SET\_TERM\_SUBVAR

SET\_PART\_OF\_TERM\_SUBVAR-FAL

SET\_PART\_OF\_TERM\_SUBVAR-SFR

SET\_PART\_OF\_TERM\_SUBVAR-SUM

ZFRROLL-CENTRAL\_CAMPUS

ZFRROLL-NORTH\_CAMPUS

ZFRROLL-SOUTH\_CAMPUS

ZSPNGSR-AUDIT\_MODE

ZSPNGSR-AUD\_COPY\_FILE

ZSPNGSR-UPDATE\_MODE

ZSPNGSR-UPD\_COPY\_FILE

ZFRROLL-ALL\_CAMPUS

SHRROLL-PT\_GRADE\_ROLL\_AUDIT

SHRROLL-PT\_GRADE\_ROLL\_UPDT

SHRRPTS-RPT\_MODE\_PT\_GRADE\_ROLL

SHRRPTS-UPD\_MODE\_PT\_GRADE\_ROLL

SHRCOPA-ALL\_PT\_GRADE\_ROLL

SHRCOPA-COLLECT\_PT\_GRADE\_ROLL

ZSPFSPU-PT\_GRADE\_ROLL

SZPTSPU-PT\_GRADE\_ROLL

ZFRRGAM-PT\_GRADE\_ROLL

REMOVE\_SUBVAR\_VALUES

General Output Prompts Predecessors Conditions Documentation

Job

Description

Alias

Eligibility

Calendar

☐ Mo ☐ Tu ☐ We ☐ Th ☐ Fr ☐ Sa ☐ Su

☐ Component Active

☐ Job Active

☐ Run ☐ Skip

Execution

Agent

Login

Output Scan

Notification

Environment Variables

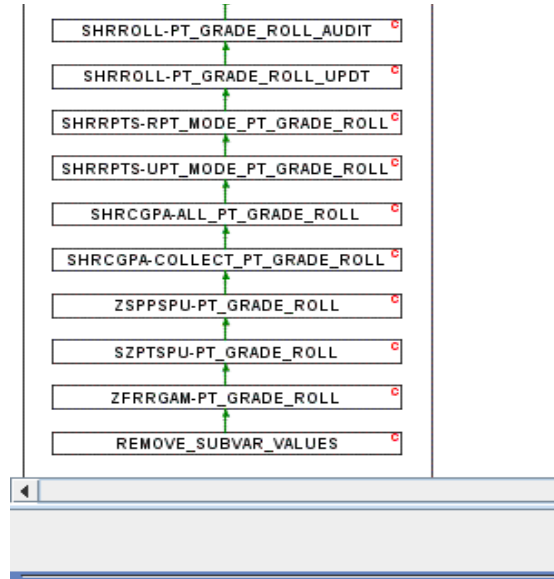
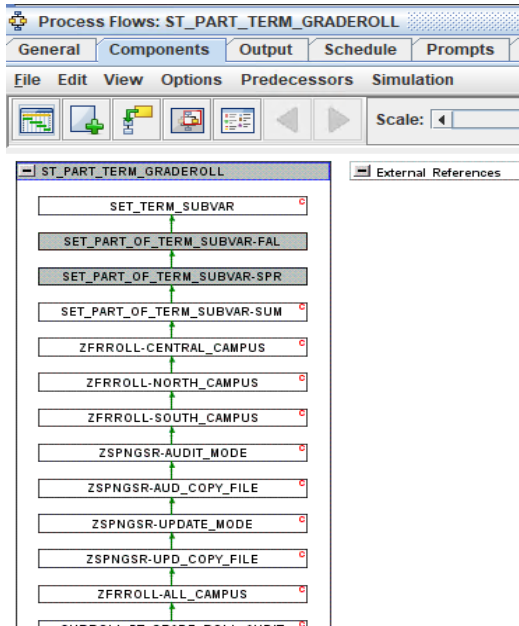
Age run time

☐ Use Request Output Device ☐ Use Job Conditions

OK Apply Request Cancel Help



# Part of Term Grade Roll



# Part of Term Grade Roll Process

- Uses a variety of different jobs and functions.
- Baseline Banner Jobs are used.
- Custom Banner Jobs are used.
- Conditions are used for Term Establishment.
- Copy file shell script is used to move files.
- Host commands are also used to call the mail-x process to send notifications.

# Part of Term Grade Roll

- Prompt Variables are used to identify the Part of Term Dates

Process Flows: ST\_PART\_TERM\_GRADEROLL


General Components Output Schedule Prompts Conditions Documentation User Groups

#	Description	
1	PART_TERM_4A_FAL_DATE	20210920
2	PART_TERM_5A_FAL_DATE	20210927
3	PART_TERM_6A_FAL_DATE	20211004
4	PART_TERM_4B6AS_FAL_DATE	20211018
5	PART_TERM_5B_FAL_DATE	20211101
6	PART_TERM_12A4C_FAL_DATE	20211115
7	PART_TERM_13B5C1W157B_FAL_DATE	20211206
8	PART_TERM_3_SPR_DATE	20220103
9	PART_TERM_4A_SPR_DATE	20220214
10	PART_TERM_5A_SPR_DATE	20220221
11	PART_TERM_6A6AS_SPR_DATE	20220301
12	PART_TERM_4B_SPR_DATE	20220321
13	PART_TERM_5B_SPR_DATE	20220404
14	PART_TERM_12A4C_SPR_DATE	20220418
15	PART_TERM_13B5C1W157B_SPR_DATE	20220509
16	PART_TERM_3_SUM_DATE	20220606
17	PART_TERM_5AMS5AT8_SUM_DATE	20220711
18	PART_TERM_8A11_SUM_DATE	20220801
19	PART_TERM_1WF1WS_SUM_DATE	20220808

Data type

Description

☐ List of Values ☐ Multi Select ☐ Comma Separated

 Copy...

# Part of Term Grade Roll

- Condition to set Term Code

SET\_TERM\_SUBVAR

Order	Timing	Performed	Condition#1	Qual	Condition#2	Action	Action Argument
1	BEFORE	CONTINUOUS	{#YYYYMMDD}	>=	20210823	SET SUBVAR	#POT_DEFINED_TERM...
2	BEFORE	CONTINUOUS	{#YYYYMMDD}	>=	20211213	SET SUBVAR	#POT_DEFINED_TERM...
3	BEFORE	CONTINUOUS	{#YYYYMMDD}	>=	20220516	SET SUBVAR	#POT_DEFINED_TERM...

**Condition Details - ST\_PART\_TERM\_GRADEROLL(SET\_TERM\_SUBVAR)**

Timing: BEFORE Condition: USER DEFINED

Test

If {#YYYYMMDD} >= 20220516

Action

Command: SET SUBVAR #POT\_DEFINED\_TERM=202230

Action Timing:

- ☐ The first time the condition is true
- ☒ Every time the condition is true
- ☐ Disabled

Buttons: New, Edit, Delete, Copy..., Up, Down

# Part of Term Grade Roll

- Condition to set Part of Term Values

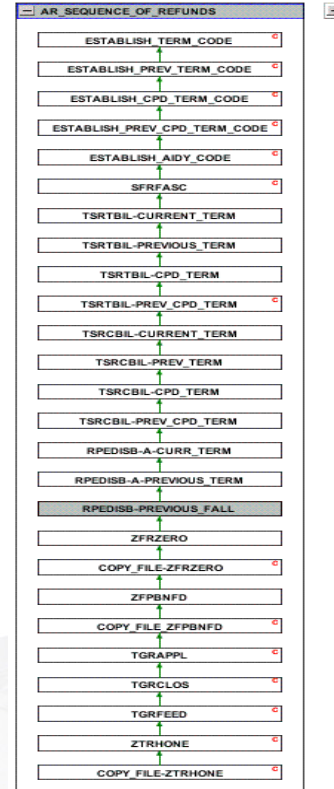
SET_PART_OF_TERM_SUBVAR-SUM								
General	Output	Prompts	Predecessors	Conditions	Documentation			
Order	Timing	Performed	Condition#1	Qual	Condition#2	Action	Action Argument	
1	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM1=3	
2	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM2=1	
3	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM3=1	
4	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM4=1	
5	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM5=1	
6	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM6=1	
7	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM7=1	
8	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM8=1	
9	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM9=1	
10	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM10=1	
11	BEFORE	CONTINUOUS	{YYYYMMDD}	=	{#16}	SET SUBVAR	#PART_OF_TERM11=1	

# Part of Term Grade Roll

- Banner baseline and custom jobs used include:
  - ZFRROLL
  - ZSPNGSR
  - SHRROLL
  - SHRRPTS
  - SHRCGPA
  - ZSPPSPU
  - SZPTSPU
  - ZFRRGAM

# Refunding Process

- Conditions establish current and previous term code.
- Baseline processes are executed.
- Custom processes are used.
- Baseline Automatic/UC4 scripts
- Custom Scripts



# Refunding Process for Finance Department

- Dates are evaluated to determine Term Code using conditions.

Order	Timing	Performed	Condition#1	Qual	Condition#2	Action	Action Argument
1	BEFORE	ONCE	{YYYYMMDD}	<=	20210913	SET SUBVAR	#AR_REFUND_TE...
2	BEFORE	ONCE	{YYYYMMDD}	<=	20220615	SET SUBVAR	#AR_REFUND_TE...
3	BEFORE	ONCE	{YYYYMMDD}	<=	20220127	SET SUBVAR	#AR_REFUND_TE...

**Condition Details - AR\_SEQUENCE\_OF\_REFUNDS(ESTABLISH\_TERM\_CODE)**

Timing: BEFORE    Condition: USER DEFINED

Test:

If {YYYYMMDD} { } <= 20220127 { }

Action:

Command: SET SUBVAR    #AR\_REFUND\_TERM\_CODE=202210 { }

Action Timing:

☒ The first time the condition is true

☐ Every time the condition is true

☐ Disabled

OK Cancel



# Refunding Process for Finance Department

- Process executed include custom and baseline using term code include:
  - SFRFASC
  - TSRTBIL
  - TSRCBIL
  - RPEDISB
  - ZFRZERO
  - TGRAPPL
  - TGRCLOS
  - TGRFEED
  - ZTRHONE

# SFRFASC – Refunding Process

SFRFASC

General Output Banner Predecessors Conditions Documentation

**Program Information**

Program: sfrfasc \*

Command Name:

Program Type: Pro\*C \*

Type Code: C \*

Login Override:

Source File Override:

**Banner Output**

Printer: NOPRINT

Printer Form:

Lines Per Inch:

**Advanced Output**

Output Search Pattern: .\*[^0-9]{ONE\_UP}[^0-9].\*

Rename Output Pattern:

**Parameters**

Prompt	Type	Description	Required	Value
01	Single	Term	Y	#AR_REFUND_TERM_CODE
02	Single	Assessment date		#sysdate
03	Single	Student ID		
04	Single	Application code		
05	Single	Selection ID		
06	Single	Creator ID		
07	Single	User ID		
08	Single	Batch type		E
09	Single	Enrollment status		%
10	Single	Full or part time indicator	Y	%
11	Single	Assessment rule type	Y	R
12	Single	Accounting detail to print	Y	C
13	Single	Report type	Y	T
14	Single	Sort order	Y	N
15	Single	Add invoice collector record	Y	N
16	Single	Invoice printer		
17	Single	Create accounting records	Y	Y
18	Single	Refund by total refund date		
19	Single	Run mode	Y	U

# TSRCBIL – Refunding Process

TSRTBIL-CURRENT\_TERM

General Output Banner Predecessors Conditions Documentation

**Program Information**

Program: tsrtbil \*

Command Name:

Program Type: Pro\*C \*

Type Code: C \*

Login Override:

Source File Override:

**Banner Output**

Printer: NOPRINT

Printer Form:

Lines Per Inch:

**Advanced Output**

Output Search Pattern: .\*[^0-9]{ONE\_UP}[^0-9].\*

Rename Output Pattern:

**Parameters**

Prompt	Type	Description	Required	Value
01	Single	Selection Identifier		
02	Single	Application Code		
03	Single	Creator Id		
04	Single	Term	Y	#AR_REFUND_TERM_CODE
05	Single	Address Selection Date	Y	#sysdate
06	Multi	Address Hierarchy	Y	1MA 2RE 3BI
07	Single	Contract ID		
08	Single	Contract Number		%
09	Single	Futures Print Date		
10	Single	Currency Code		
11	Single	Run Mode	Y	APPLYCRED
12	Single	Apply Third Party Deposits Ind	Y	Y
13	Single	Print Future Detail Indicator	Y	Y
14	Single	Print Order	Y	N
15	Single	Billing Date	Y	#sysdate
16	Single	Days in Billing Cycle	Y	30
17	Single	Minimum Acct Balance	Y	-9999999.99
18	Single	Detail Codes to Bill(% valid)	Y	%
19	Single	Affect Amount Due Indicator	Y	Y
20	Single	Print Only Required Course	Y	N

# Refunding Process for Finance Department

- Each process uses conditions to send notification to users.
- Output is included in notifications.
- Notifications can be custom made with:
  - Substitution Variables
  - Internal Job ID variables.
  - Custom Text
  - Standard messages.

# TSRCBIL – Notification Process

TSRIBIL-CURRENT\_TERM

Order	Timing	Performed	Condition#1	Qual	Condition#2	
1	AFTER	ONCE	{status}	=	FINISHED	SEND N
2	AFTER	ONCE	{status}	!=	FINISHED	SEND N

**Condition Details - AR\_SEQUENCE\_OF\_REFUNDS(TSRIBIL-CURRENT\_TERM)**

Timing: AFTER Condition: USER DEFINED

Test:

If {status} {} = {} FINISHED {}

Action:

Command: SEND NOTIFICATION AR\_TSRIBIL\_NOTIFY F Details

Action Timing:

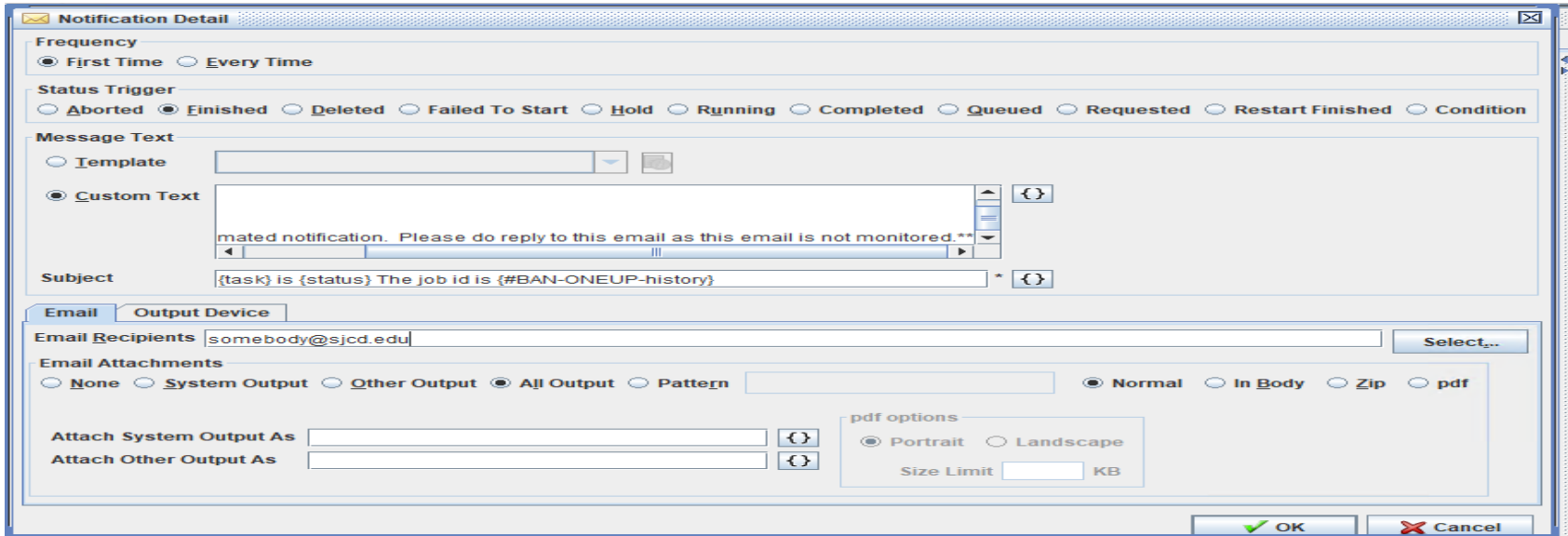
☒ The first time the condition is true

☐ Every time the condition is true

☐ Disabled

OK Cancel

# TSRCBIL – Notification Process



The screenshot shows a "Notification Detail" dialog box with the following sections:

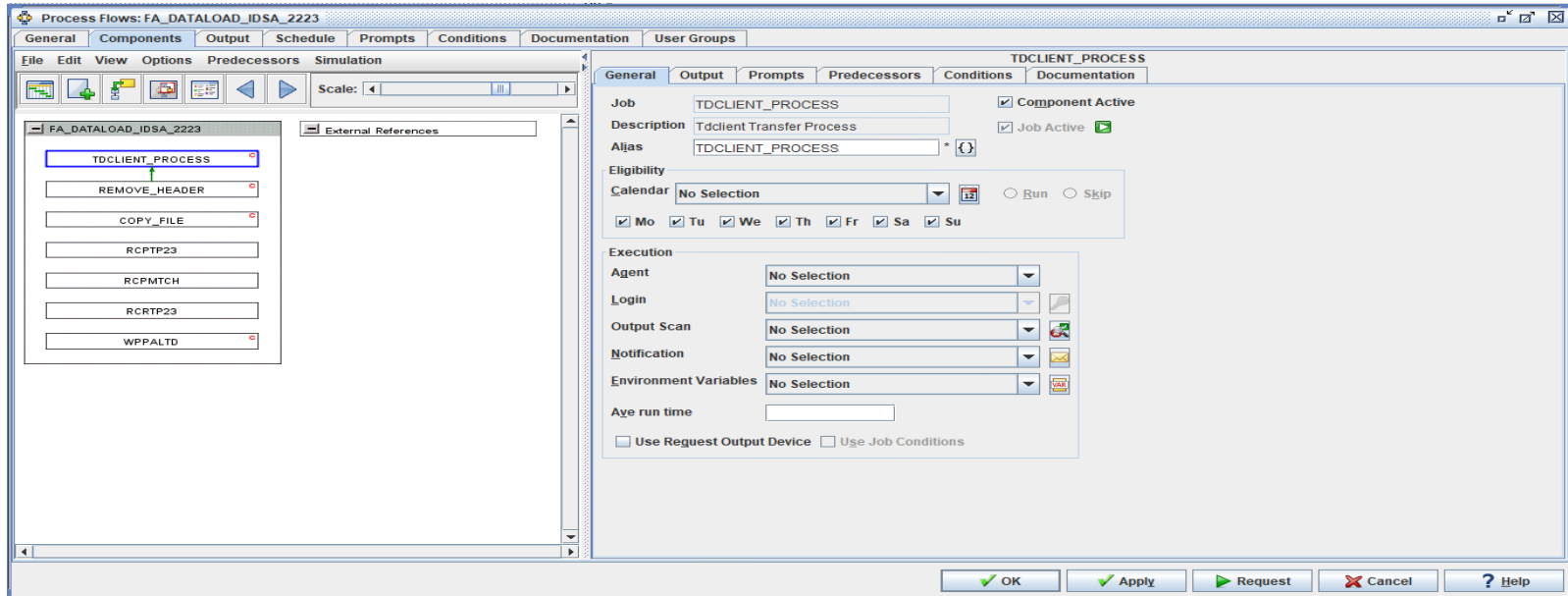
- Frequency:** ☒ First Time, ☐ Every Time
- Status Trigger:** ☐ Aborted, ☒ Finished, ☐ Deleted, ☐ Failed To Start, ☐ Hold, ☐ Running, ☐ Completed, ☐ Queued, ☐ Requested, ☐ Restart Finished, ☐ Condition
- Message Text:**
  - ☐ Template (with a dropdown menu)
  - ☒ Custom Text (with a text area containing "mated notification. Please do reply to this email as this email is not monitored.\*\*")
- Subject:** {task} is {status} The job id is {#BAN-ONEUP-history} \* (with a dropdown menu)
- Email / Output Device:**
  - Email Recipients:** somebody@sjcd.edu (with a "Select..." button)
  - Email Attachments:** ☐ None, ☐ System Output, ☐ Other Output, ☒ All Output, ☐ Pattern (with a dropdown menu)
  - pdf options:** ☒ Normal, ☐ In Body, ☐ Zip, ☐ pdf; ☒ Portrait, ☐ Landscape; Size Limit [ ] KB
  - Attach System Output As:** [ ] (with a dropdown menu)
  - Attach Other Output As:** [ ] (with a dropdown menu)

Buttons at the bottom:  OK,  Cancel

# Financial Aid Dataload Process

- Process to load the ISIR Files such as:
  - IDSAXXOP
  - IGSGXXOP
  - IGDSXXOP
  - IGCXXOP
- Uses Custom script to automate the download of ISIR files.
- Custom scripts are used to manipulate files by removing header and trailer records.
- Process is scheduled to run daily.

# Financial Aid Dataload Process





# Financial Aid Dataload Process

- TDClient script is used to specify the exact message class to download.
- Additional scripts are used to remove header and trailer records.
- Copy file scripts are used to load file to dataload directory.
- Banner jobs to handle the dataload procedures are executed.
- All output is emailed to financial aid staff.

# Financial Aid Dataload Process

TDCLIENT\_PROCESS

General

Output

Prompts

Predecessors

Conditions

Documentation

#	Description	Default	Value	Re...
1	Reset Option	Y	Y	Y
2	Transmission Command	T	R	Y
3	Path & File Name		{#TDClient_Directory}/incoming/{#...	Y
4	Message Class		{#3}	Y
5	TG Destination Number		TG53689	Y

Select...

{ }

# TDClient Script variables

TDCLIENT\_PROCESS

General

Output

Prompts

Predecessors

Conditions

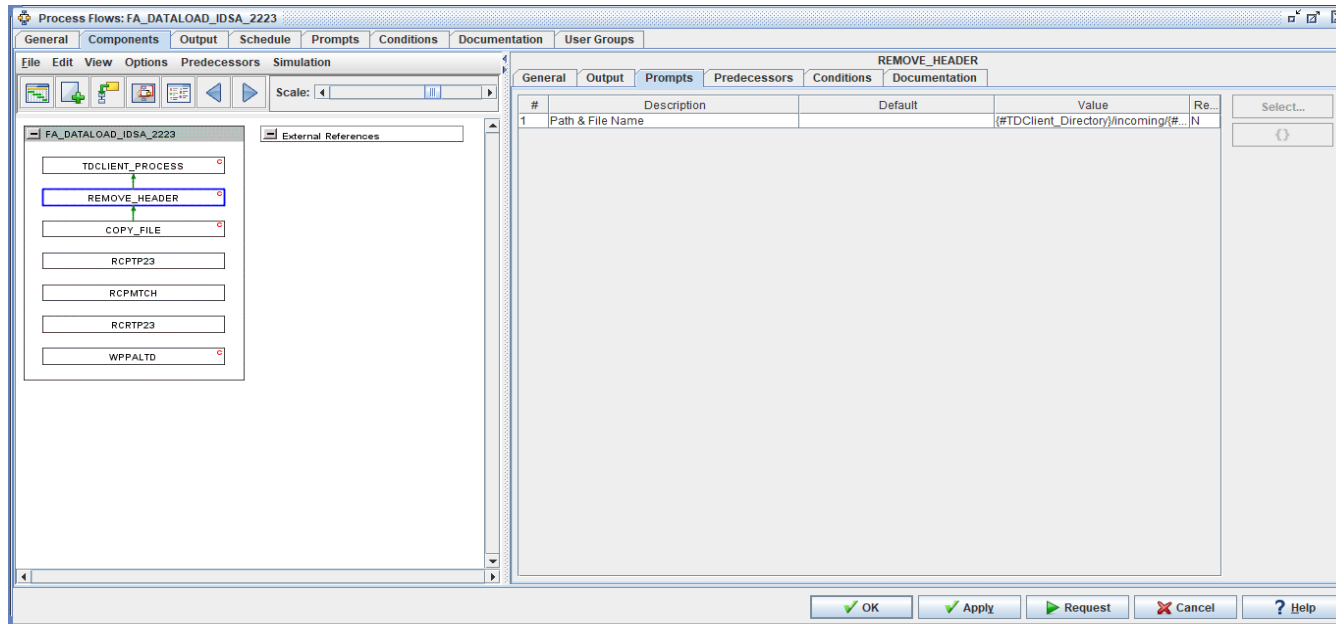
Documentation

#	Description	Default	Value	Re...
1	Reset Option	Y	Y	Y
2	Transmission Command	T	R	Y
3	Path & File Name		{#TDClient_Directory}/incoming/{#...	Y
4	Message Class		{#3}	Y
5	TG Destination Number		TG53689	Y

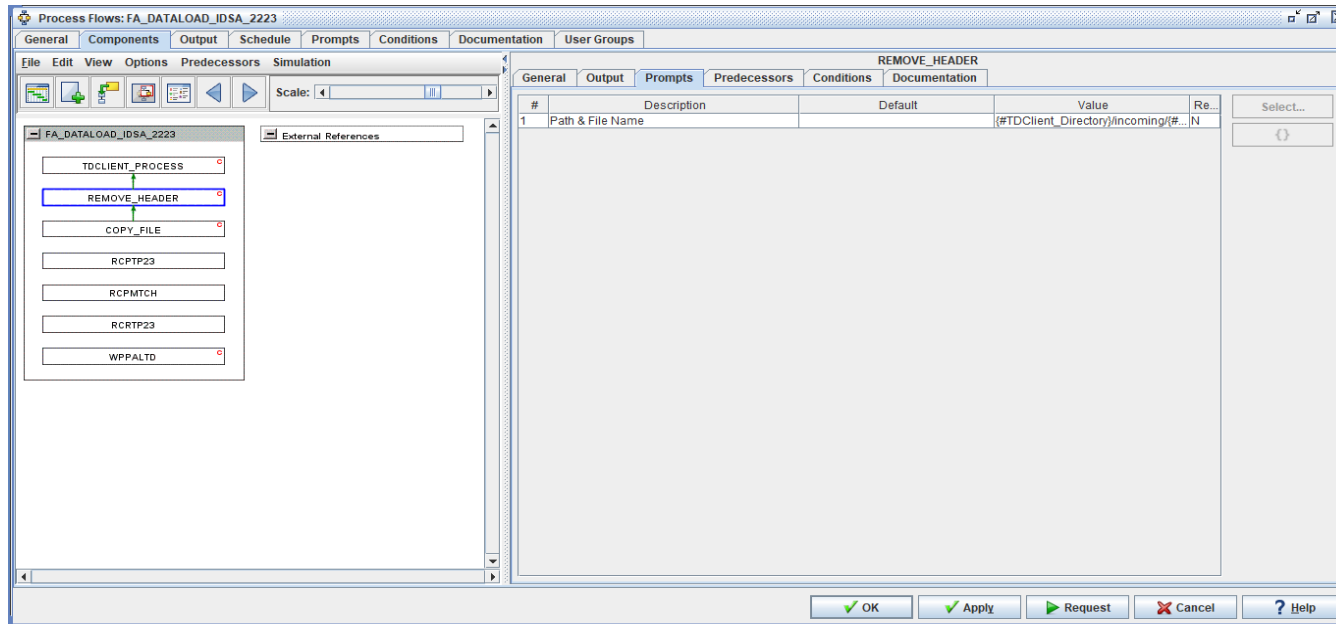
Select...

{ }

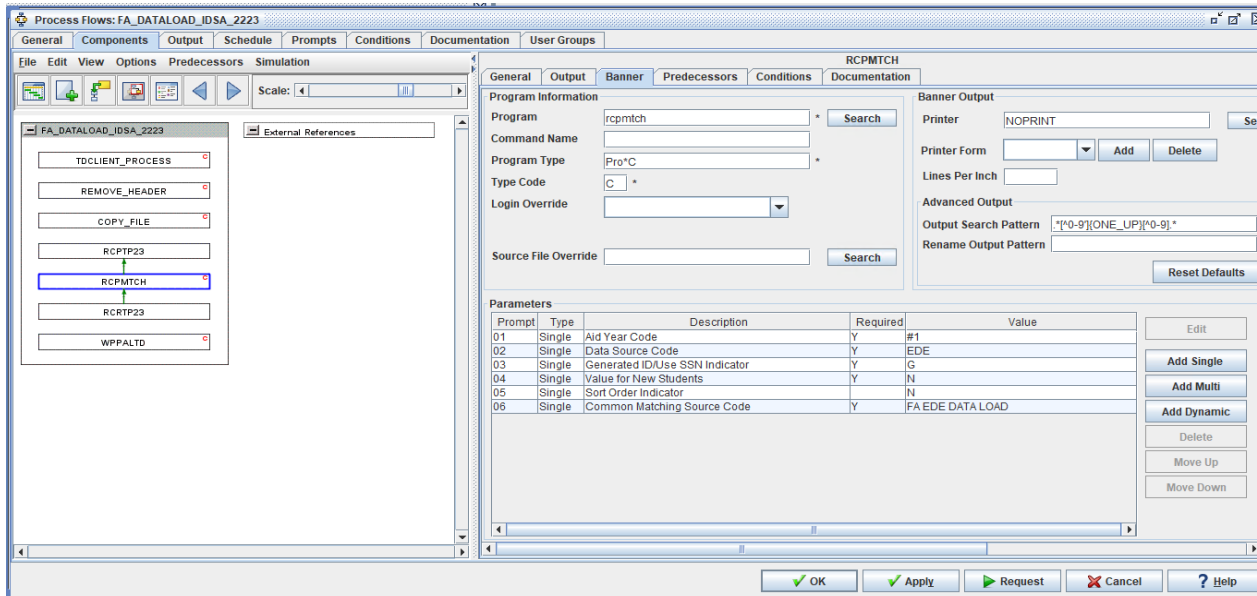
# Remove Header Script parameters



# Remove Header Script parameters



# RCPMTCH parameters



Process Flows: FA\_DATALOAD\_IDSA\_2223

General Components Output Schedule Prompts Conditions Documentation User Groups

File Edit View Options Predecessors Simulation

Scale: 1

FA\_DATALOAD\_IDSA\_2223

External References

TDCLIENT\_PROCESS

REMOVE\_HEADER

COPY\_FILE

RCPTP23

**RCPMTCH**

RCRTP23

WPPALTD

RCPMTCH

General Output Banner Predecessors Conditions Documentation

Program Information

Program rcpmtch

Command Name

Program Type Pro\*C

Type Code C

Login Override

Source File Override

Banner Output

Printer NOPRINT

Printer Form

Lines Per Inch

Advanced Output

Output Search Pattern \*["0-9"]{ONE\_UP}["0-9"]\*

Rename Output Pattern

Reset Defaults

Parameters

Prompt	Type	Description	Required	Value
01	Single	Aid Year Code	Y	#1
02	Single	Data Source Code	Y	EDE
03	Single	Generated ID/Use SSN Indicator	Y	G
04	Single	Value for New Students	Y	N
05	Single	Sort Order Indicator	N	N
06	Single	Common Matching Source Code	Y	FA EDE DATA LOAD

Edit

Add Single

Add Multi

Add Dynamic

Delete

Move Up

Move Down

OK Apply Request Cancel Help

Thank you for attending



# Contact

Mark Deaver

- Business Analyst II – San Jacinto
- [mark.deaver@sjcd.edu](mailto:mark.deaver@sjcd.edu)

