

BAM ↔ Banner Data Exchange Architecture and Process Overview

Contents

- 1. Introduction.....2
- 2. General Overview of the BAM→Banner→BAM Workflow.....2
- 3. General Terms4
- 4. Background and System-of-Record Rationale4
- 5. Required Technical Components and Skills5
- 6. Keys (Technical Identifiers & Configuration Elements).....6
- 7. BAM Setup.....6
- 8. Program 1 — Extract Current Offers from BAM.....9
- 9. Program 2 — Load BAM Data into the Custom Banner Table 13
- 10. Program 3 — Insert and Update Annual Awards in Banner 17
- 11. Program 4 — Insert and Update Term-Based Awards in Banner.....21
- 12. Program 5 — Extract Banner Award Decisions for Return to Blackbaud Awards Management..... 26
- 13. Program 6 — Import Banner Award Decisions Back into BAM..... 30
- 14. Program 7 — Report Distribution to MS Teams..... 31
- 15. Other Institutional Processes 31
- BAM ↔ Banner Integration – Full Technical Document..... 33
- Summary 34
- Conclusion..... 34
- Glossary of Terms 34
- 16. Contact Information 35

1. Introduction

This document provides a comprehensive technical overview of the data-exchange architecture between **Blackbaud Award Management (BAM)** and **Ellucian Banner** as implemented at the College of Charleston. Here at the College of Charleston, the marriage of these two applications is referred to as the Cougar Scholarship Award System, or CSAS. This interface was built in accordance with the business rules of our Financial Aid Office, Institutional Advancement, and Information Technology Departments.

Its purpose is to outline the end-to-end workflow, data structures, interface programs, and validation logic used to synchronize scholarship award information between the two systems. The document emphasizes **functional flow**, **interface dependencies**, and **processing requirements**, but does not include source code.

Questions, clarifications, or enhancement requests may be directed to the author via email. (Swindellvc@charleston.edu)

2. General Overview of the BAM→Banner→BAM Workflow

The BAM–Banner integration operates as a cyclic exchange of award information. The high-level workflow consists of two primary operations:

1. **Import into Banner:**

Export all scholarships in an **Offered** status from BAM, then load and process them into Banner using custom staging tables and Banner award APIs.

2. **Return Banner Decisions to BAM:**

Identify Banner-stored BAM awards whose decisions (**Accepted**, **Declined**, **Canceled**) have changed within the past seven days, and transmit the updated statuses back to BAM. The period is flexible and can be any time period you choose.

2.1 High-Level Process Summary (Technical Table)

Stage	System	Description
1. Extract Current Offers	BAM → Server	Use the Fund Disbursement API to export all awards in “Offered” status for the selected aid year/period.
2. Load Files into Staging Table	Server → Banner	Parse exported BAM files and load records into our custom table custom.cofc_csas_awards .
3. Insert/Update Annual Awards	Banner	Evaluate and process annual awards using the RPRAWRD Banner API. (BANINST1 PACKAGES)
4. Insert/Update Term Awards	Banner	Process term-based awards through Banner APIs (RPRATRM).
5. Extract Banner Decision Updates	Banner → Server	Identify Banner awards with decision changes within X days (default: 7).
6. Send Updates Back to BAM	Server → BAM	Upload formatted CSV to BAM via the Fund Disbursements Updater API.
7. Reporting	Server → Teams	Store transaction logs and reports in Teams for reconciliation.

3. General Terms

The following terms are used throughout this integration:

3.1 Definitions (Table Format)

Term	Definition
Academic Works	The legacy platform was originally used for institutional scholarship management. Blackbaud acquired Academic Works on April 3, 2017 .
Blackbaud Award Management (BAM)	The current name and platform, replacing Academic Works, provide expanded award processing and administrative features.
Banner	An Ellucian ERP used for student and financial aid records, functioning as the system of record for authoritative award decisions.

4. Background and System-of-Record Rationale

Banner is designated as the **system of record** for all scholarship awards. This architectural decision ensured that all award type updates—including federal, state, local, institutional, and external funds—remained centralized and authoritative.

BAM allows students to accept awards within its UI, but BAM's date-handling and restricted filtering/reporting make it difficult to reliably determine the timeline of decision changes. Award statuses may appear inconsistent between the two systems (e.g., **Accepted in BAM** but **Declined in Banner**).

To prevent these discrepancies:

- Only **offer** records flow from BAM → Banner.
- All **accept/decline/cancel** decisions are made exclusively in Banner. (RTVAWST_CODE)
- A nightly job extracts Banner updates and imports them back to BAM to keep BAM consistent, using the BAM API.

This design ensures consistent award reporting and prevents data divergence.

5. Required Technical Components and Skills

The BAM ↔ Banner integration requires a combination of scripting capability, API usage, Banner financial-aid domain expertise, and controlled server execution environments. The following components are foundational to the successful operation and maintenance of the interface:

Component	Description
PL/SQL or Equivalent Scripting Language	Required to develop automated procedures executed through the UC4 job scheduler.
Knowledge of cURL	cURL (short for "Client URL") is a command-line tool and software library used by programmers to transfer data to and from a server. At its core, it allows you to interact with internet resources directly from your computer's terminal or within a software script by typing a URL and specifying what data you want to send or receive
Job Scheduler	An enterprise job-scheduling system is used to orchestrate scripts, batch processes, and interface programs. We use ATOMIC UC4
Local Server Environment	A secure environment capable of executing cURL commands to interact with BAM APIs.
Ellucian Banner	ERP system providing APIs, data structures, and award-processing tables used by the integration.
Banner Development Expertise	Required to build and maintain PL/SQL packages, APIs, validations, and custom schemas, such as custom.cofc_csas_awards .
BAM API Documentation	Found under Site → Settings → Import and Export Tools , API Documentation Link at the bottom, providing details on Fund Disbursement and Updater API specifications.
CRON Server	is a computer or service that runs the cron daemon, a utility in Unix-like operating systems used for scheduling tasks (called "cron jobs") to run automatically at specified dates and times. These tasks are defined using specific syntax in a configuration file called a crontab. Runs the API scripts.

These components collectively support API communication, data validation, controlled scheduling, secure file exchange, and synchronization workflows across institutional systems.

6. Keys (Technical Identifiers & Configuration Elements)

BAM and Banner rely on specific identifiers, network paths, and tokens to ensure secure transfers and reliable data mappings. Key elements include:

Key Item	Description
NET_WORK_DRIVE	The institution-managed a network directory where interface files are generated and processed. Must be accessible by UC4 and CRON SERVER
Institution BAM URL	Format: <i>yourschool.academicworks.com</i> . Represents the base endpoint for all API interactions.
Token Key	Authentication token used for all BAM API transactions, obtained from <i>Site → Settings → Import & Export Tools → Import Token</i> .
Output Files (csas_file01.txt, etc.)	Files generated from the Fund Disbursement API containing student, fund, decision, and award-period metadata.

These identifiers must be kept secure and updated yearly in accordance with BAM and Banner aid-year structures.

7. BAM Setup

Proper BAM configuration ensures that award periods, categories, and term alignments accurately map to Banner’s aid-year and term structure.

7.1 Fund Period Alignment

BAM fund periods mirror the institution’s Banner aid-year configuration (e.g., **2728** → *Fall 2027/Spring 2028*). This alignment ensures that exported award **offers** correspond to the correct Banner award cycles. RPTAWRD_AIDY_CODE

BAM fund periods follow the same sequential format (e.g., **2526, 2627, 2728**, etc.).

Term-Based Fund Period Format

BAM term periods follow the naming convention:

BANNER_TERM – AID_YEAR – DESCRIPTION

Example:

202710-2627-Fall 2026

This identifies a Fall 2026 term-based award record.

7.2 Application Categories

Application Categories are maintained under:

Site → Configuration → Application Categories

Each category has an internal numerical identifier used during data translation to Banner.

Example from source:

Selecting [EDIT] the **“Offered”** category may display a URL such as:

.../admin/categories/7/edit

indicating that the category ID for **Offered** is **7** in our configuration

Award Management Site Opportunity Stewardship

Name	Visible To	Application Impact	Award Effects	Applications	
Drafted	Applicants	The application has been started but not yet submitted to any opportunity.	None	1527	Edit
Submitted	Applicants and Reviewers	The application is ready for consideration by staff or reviewers.	None	15545	Edit
Declined	Applicants and Reviewers	An award has been rejected and the application process is completed for that applicant.	None	2	Edit
Offered	Applicants and Reviewers	An award has been offered to an applicant, and he or she can now accept or reject this offer.	Encumbers Funds	9	Edit
Accepted	Applicants and Reviewers	An award has been accepted and the application process is completed for that applicant.	Encumbers Funds	377	Edit
Declined - Admin	Administrators Only	None	None	9	Edit
Finalist	Reviewers	None	Encumbers Funds	None	Edit
Alternate	Reviewers	None	None	None	Edit
Disabled					
Finalist	Applicants and Reviewers	None	None	None	Edit
Alternate	Reviewers	None	Encumbers Funds	None	Edit

Automatch Categories

Name	Visible To	Application Impact	Award Effects	Applications	
Submitted	Reviewers	The application is ready for consideration by staff or reviewers.	None	--	Edit
Declined	Applicants and Reviewers	An award has been rejected and the application process is completed for that applicant.	None	None	Edit
Offered	Applicants and Reviewers	An award has been offered to an applicant, and he or she can now accept or reject this offer.	Encumbers Funds	22	Edit
Accepted	Applicants and Reviewers	An award has been accepted and the application process is completed for that applicant.	Encumbers Funds	1197	Edit

7.3 GET Current Offers (Preparation for Program 1)

The BAM Fund Disbursement API is used to extract all scholarships assigned to the **Offered** category for a selected aid year and term. Program 1 retrieves these offers and writes them to files for downstream Banner processing.

The current BAM API request supports **up to 1,000 records**; additional output files may be generated using pagination parameters (e.g., page=2).

The exported files include the following fields (in order):

BAM Field	Description
award_period_name	Aid year period representing academic year or term you are processing (2627). You can include different aid years in each call if you are running more than one term like fall and summer.
user_uid	Banner User ID (SATURN.SPRIDEN_ID).
portfolio_code	Banner Fund Code (RFRBASE.FUND_CODE).
disbursement_amount	Award amount.
initial_application_category_id	BAM decision category ID (e.g., Offered).
Id	BAM Opportunity ID.
award_period_id	Internal BAM period identifier. Awards. Periods are defined in Site -> Configuration -> Award Period
form_field_10473	An indicator of student active status.

These fields become the initial dataset for Program 2, where Banner validations begin.

8. Program 1 — Extract Current Offers from BAM

Program 1 is responsible for extracting all scholarship opportunities in the **Offered** category from Blackbaud Award Management (BAM). The script uses the **Fund Disbursement API**, available under **BBAM → Site → Settings → Import and Export Tools**. It writes the resulting data to one or more output files for downstream Banner processing.

This program is executed on a secure institutional server and is typically scheduled to run prior to Program 2 in the UC4 job scheduler environment.

8.1 API Extraction Logic

The current Fund Disbursement API supports **up to 1,000 records per request**. If more than 1,000 records exist for the specified term/aid year, the script uses **pagination** (via the `page=` parameter) to generate additional output files. This allows the process to scale for large awarding cycles without modification of the core logic.

The script sends output to the institution's network drive (**NET_WORK_DRIVE**) and produces file names such as:

- `csas_file01.txt`
- `csas_file02.txt`
- *...additional files as needed*

NOTE: This is the file naming of our choosing

8.2 Example API Call

The following represents the structure used by Program 1 to query BAM. This is a direct reflection of the program's functional behavior as described in your source:

```
curl --verbose -g -o "\\NETWORK_DRIVE\csas_file01.txt"  
"https://yourschool.academicworks.com/api/v1/disbursements.csv?c[]=initial_application  
_category_name__contains__Offered&c[]=opportunity_archived__eq__0&c[]=award_perio  
d_name__contains__2627&page=1&per_page=1000&fields=award_period_name,user_uid,  
portfolio_code,disbursement_amount,initial_application_category_id,id,award_period_id,f  
orm_field_10473&token=tokenkey"
```

This reflects the parameters listed in the API document, including filtering by:

- application category (Offered)
- non-archived status
- award period (aid year)
- pagination via page=n
- record limit (per_page=1000)
- explicit field selection

Note: highlights indicate that award period values must be updated each aid year. If multiple aid years or terms need to be processed in the same run, multiple API calls may be executed.

Note we just created a power shell code this year (2026) that can automatically get this data for a span of terms.

8.3 Award Period Configuration Dependency

Award periods are configured in:

Site → Configuration → Award Periods

These values **must** match Banner aid-year structures for correct data alignment.

THE BAM AWARDS PERIOD TABLE

Award Periods

Existing Award Periods 

Name	API ID	Disabled?
2526	81	<input type="checkbox"/>
202610-2526-Fall 2025	82	<input type="checkbox"/>
202620-2526-Spring 2026	83	<input type="checkbox"/>
202630-2526-Summer 2026	84	<input type="checkbox"/>
2627	114	<input type="checkbox"/>
202710-2627-Fall 2026	115	<input type="checkbox"/>
202720-2627-Spring 2027	116	<input type="checkbox"/>
202730-2627-Summer 2027	117	<input type="checkbox"/>
1920	1	<input checked="" type="checkbox"/>
2021	2	<input checked="" type="checkbox"/>
2122	3	<input checked="" type="checkbox"/>
202310-2223-Fall 2022	4	<input checked="" type="checkbox"/>

8.4 Output File Data Structure

The BAM .csv or .txt output files created in Program 1 contain the following fields in the specified order:

Field Name	Description	Mapping
award_period_name	Aid year or term of the award period	Award cycle identifier AID YR-- SEE: : SITE->CONFIGURATION->AWARD PERIOD
user_uid	Banner User ID (SPRIDEN_ID)	Student identifier
portfolio_code	Banner Fund Code (RFRBASE.FUND_CODE)	Scholarship fund
disbursement_amount	Award offer amount	Currency value
initial_application_category_id	BAM category ID (e.g., Offered = 7)	Decision code (Offer)
id	BAM Opportunity ID	Unique award identifier
award_period_id	Internal BAM period ID	Period metadata SEE: SITE->CONFIGURATION->AWARD PERIOD
form_field_10473	Boolean field indicating student active status	Used for validation; inactive students skipped

These fields form the input to Program 2, which loads the data into a custom Banner staging table.

8.5 Scheduling & Execution Environment

Program 1:

- Runs on a secure server configured for cURL communications with BAM.
- Requires network access to the institution's BAM domain (*.academicworks.com).

- Outputs files to an accessible [network directory](#).
 - Is scheduled to execute prior to Program 2, often daily or multiple times per week, depending on awarding cycles.
-

8.6 Key Notes & Operational Constraints

- Award period names **can be updated annually (manually) or automatically annually** in the script.
- Multiple aid years/terms may be combined by running multiple API calls.
- File naming conventions (csas_file01.txt, etc.) allow unlimited scalability.
- Network security must allow outbound HTTPS communication to the BAM API.

9. Program 2 — Load BAM Data into the Custom Banner Table

Program 2 processes the files generated by Program 1 and imports the extracted BAM data into the custom Banner staging table **custom.cofc_csas_awards**. This staging table acts as the authoritative temporary repository for all pending BAM → Banner award records.

The program runs on the institutional server used by **UC4**, typically **10 minutes after Program 1**

9.1 Functional Overview

Program 2 performs the following core operations:

1. **Clear the staging table** (custom.cofc_csas_awards).
2. **Read each BAM file** created by Program 1 (csas_file01.txt, csas_file02.txt, etc.).
3. **Validate each record** using several Banner-specific rules.
4. **Insert valid rows** into the custom staging table.
5. **Skip records** where the BAM “active student” indicator (form_field_10473) reflects an inactive status.
6. **Construct the proprietary BBAM_ID key** used for mapping award records back to BAM.

9.2 BBAM_ID Construction

A unique linking identifier called **BBAM_ID** is generated and stored in Banner to ensure that:

- Banner can identify awards originating from BAM
- Banner can return acceptance/decline/cancel updates back to BAM
- Duplicate imports can be safely prevented

Format: 'BBAM_' + <id> + '_' + <termID>

Where:

- <id> = BAM Opportunity ID
- <termID> = Banner or BAM award period term identifier

example: BBAM_1221848_113

Important: If the **BBAM_ID** is overwritten inside Banner, the award can no longer be matched to its corresponding record in BAM. This will break return-file generation.

9.3 UC4 Job Information

Job Name	Description
SZP_READ_BBAM_FILES	Executes Program 2 and loads the parsed BAM award offer data into the Banner staging table.

You may rename the job in accordance with institutional standards.

9.4 Staging Table Structure

The custom Banner table **custom.cofc_csas_awards** contains the following fields, each directly mapped from the BAM export:

Field Name	Description
IA_AIDY	Banner Aid Year
IA_ID	Student CWID
IA_FUND	Banner Fund Code (must match BAM portfolio_code)
IA_AMT	Award Amount
BAM_ID	BAM Transaction ID (required for returning decisions to BAM)

DEC_CD	BAM Decision Code (Offer)
---------------	---------------------------

These values populate the annual or term award-processing logic in Programs 3 and 4.

9.5 Annual vs. Term-Based Award Handling

The program identifies annual vs. term-based awards based on the structure of the award_period_name:

Award Type	Indicator in award_period_name	Example
Annual Awards	<i>No hyphen (-)</i>	2526,2627,2728
Term-Based Awards	<i>Contains a hyphen (-)</i>	202720-2627-Spring 2027

Records are routed accordingly in later processing steps.

9.6 Validation Logic

Program 2 performs several validation checks **before** inserting any record. These validations prevent data integrity issues downstream.

Validation	Purpose
Aid Year Length Check	Ensures IA_AIDY conforms to Banner aid-year standards.
Active Student Check (form_field_10473)	Skips inactive student records from BAM.
Student Existence in Banner	Ensures the student exists in Banner; prevents importing orphan records.
Fund Code Length Check	Ensures fund value matches the expected Banner structure.
Award Amount Present	Skips rows missing award amounts.
Decision Code Exists	Ensures decision code is valid for Banner translation.

If data appears across multiple files, the validation logic is applied to each file.

9.7 Data Flow Summary for Program 2

Step	Operation
1	Clear existing staging table data
2	Load first BAM output file
3	Validate each record
4	Insert valid records into custom.cofc_csas_awards
5	Skip invalid or inactive student records
6	Generate or update BBAM_ID
7	Repeat steps 2–6 for all remaining files
8	Commit the transaction

This prepares all offer data for consumption by **Program 3 (Annual Awards)** and **Program 4 (Term Awards)**.

10. Program 3 — Insert and Update Annual Awards in Banner

Program 3 reads annual-based award records from the [custom.cofc_csas_awards](#) staging table populated in Program 2 and processes them into Banner’s financial-aid award structure. The program determines whether each award should be **inserted**, **updated**, or **skipped**, and produces a detailed transaction log for audit and reconciliation purposes.

Records processed by this program are strictly **annual awards**, identified by an award_period_name value that **does not contain a hyphen (“-”)**.

Processing occurs using Banner’s **RPRAWRD** API located in BANINST1.PACKAGES.RP_AWARD (FAISMGR.RPRAWRD).

10.1 Core Functional Responsibilities

Program 3 performs the following operations for each annual award record:

1. Reads all staged records from [custom.cofc_csas_awards](#).
 2. Validates each record based on Banner and institutional business rules.
 3. Translates BAM decision codes into Banner award status (AWST) codes.
 4. Determines if the award record already exists in Banner.
 5. Creates new annual award records or updates existing ones.
 6. Logs all operations and results to a transaction report.
-

10.2 Eligibility and Validation Rules

The program applies the following validation criteria before inserting or updating an annual award:

Validation	Description
Fund Code Exists	Confirms the Banner fund exists in FAISMGR.RFRBase .
Annual Award Check	Ensures the award is annual — award_period_name must not contain “-”.

Decision Code Translation	Converts BAM decision codes into Banner's AWST codes. (Currently, all imported decisions are Offers.)
Record Existence Check	Determines if the award already exists in Banner using BBAM_ID and the Banner API rp_award.f_exists.

These validation rules ensure that only valid, properly structured, and compatible annual awards enter Banner.

10.3 Decision Code Translation Matrix

Program 3 uses a translation matrix to map BAM decision codes to Banner AWST decision codes. Per institutional configuration, all incoming decision types map to **Offered**.

BAM Code	Description	Banner AWST Code	Banner Meaning
02	Offered (Auto Match)	O	Offered
07	Offered (Apply)	O	Offered

10.4 Determining Record Existence in Banner

The program checks for an existing Banner award using:

- **RPRAWRD_DATA_ORIGIN** (contains the unique BBAM_ID)
- **Banner API:** rp_award.f_exists

If either check indicates the award already exists, the program routes the record for **update** instead of creation.

10.5 Processing Logic

A. Creating a New Annual Award

If no existing record is found, the program creates a new award using rp_award.p_create. Parameter values are populated as follows:

API Parameter	Value Source

p_aidy_code	IA_AIDY
p_pidm	Banner PIDM resolved from IA_ID
p_fund_code	IA_FUND
p_awst_code	Translated AWST code (Offer = O)
p_offer_amt	IA_AMT
p_offer_date	System date
p_accept_amt	IA_AMT (if applicable)
p_accept_date	System date
p_data_origin	BBAM_ID

All values originate from the staging table or runtime translation.

B. Updating an Existing Annual Award

If the award exists and the **award amount differs**, the program updates the record using rp_award.p_update.

Key characteristics of updates:

- Award amount is replaced with the value from BAM.
- The BBAM_ID is verified; if absent, it is added.
- Only OFFER records are updated; other statuses are not processed.

Note: If an administrator manually changes an offer amount in Banner, it will be overwritten by the BAM offer amount during the next import, because data flow for offers is one-directional (BAM → Banner).

C. Skipping a Record

A record is skipped if:

- The award exists and no amount change is required.
 - Validation rules fail (fund missing, invalid aid year, etc.).
 - The record is not an annual award.
-

10.6 Transaction Reporting

Program 3 generates a transaction report summarizing every processed award. The report includes:

- Student CWID
- Student Email
- Student Name
- Aid Year
- Fund Code
- Amount
- BAM Code (translated)
- Banner AWST Code
- Operation (Added, Updated, Skipped)
- Outcome Status

Our process copies the reports that are stored on a designated network location and automatically copies them to the **MS Teams Scholarship folder** through a separate process. Reports are retained for **two months**.

10.7 Scheduling Details

The script:

- Resides on the same server used by UC4 for scheduled execution
- Runs immediately after Program 2
- Institutional UC4 job name: **SZP_IMPORT_BBAM_DATA**

11. Program 4 — Insert and Update Term-Based Awards in Banner

Program 4 performs the same core functions as Program 3 but applies them exclusively to **term-based awards**. It processes all staging-table records whose `award_period_name` **contains a hyphen (“-”)**, which indicates a term-specific award structure (e.g., 202210-2122-Fall 2021).

The program reads from the `custom.cofc_csas_awards` table, determines the correct action (insert, update, or skip), and logs all outcomes for auditing.

Processing is performed using Banner’s term-based award API **RPRATRM**, accessed through `BANINST1.PACKAGES.RP_AWARD_SCHEDULE`.

11.1 Core Functional Responsibilities

Program 4:

1. Reads all **term-based** staged records.
 2. Validates the data based on Banner’s term-award requirements.
 3. Translates BAM decision codes into Banner AWST codes.
 4. Determines whether a term award exists in Banner.
 5. Creates or updates the term award accordingly.
 6. Generate a transaction report for reconciliation.
-

11.2 Term-Based Award Identification

A record is considered **term-based** if:

- `award_period_name` **contains a hyphen (“-”)**
- The term-period format conforms to:

TERM - AIDYR-DESCRIPTION

Example: 202210-2122-Fall 2021

11.3 Validation Logic

The following validations occur before any Banner processing:

Validation	Description
Fund Code Exists	Confirms the Banner fund exists in RFRBASE.
Term-Based Award Check	Award period name must contain “-” .
Term Format Validation	Must match TERM-AIDYR-DESC format.
Decision Code Translation	Converts BAM codes to Banner AWST codes.
Record Existence Check	Uses BBAM_ID or rp_award_schedule.f_exists to detect preexisting term awards.

11.4 Decision Code Translation Matrix

BAM Code	Description	Banner AWST Code	Banner Meaning
02	Offered (Auto Match)	O	Offered
07	Offered (Apply)	O	Offered

All term-based decisions imported through this interface are treated as **Offers**.

11.5 Determining Record Existence

A term-based award is considered existing if:

- A matching **BBAM_ID** exists in Banner’s RPRAWRD or RPRATRM structures, or
- Banner API rp_award_schedule.f_exists returns Y

If the award exists → the program executes an **update**.
If not → the program performs an **insert**.

11.6 Processing Logic

A. Creating a New Term Award

If no existing term award is found, Program 4 calls:

rp_award_schedule.p_create

Key parameter mappings include:

- p_aidy_code → IA_AIDY
- p_pidm → Banner PIDM resolved from IA_ID
- p_fund_code → IA_FUND
- p_period → Extracted award period
- p_term_code → Extracted Banner term code
- p_offer_amt → IA_AMT
- p_offer_date → System date
- p_awst_code → Translated Offer code (“O”)
- Other override and indicator fields set per institutional defaults

All values originate from Program 2’s staging table and system logic.

B. Updating an Existing Term Award

If the term-based offer exists and the award amount is different, the program calls:

rp_award_schedule.p_update

Update behavior:

- Replaces the award amount with the BAM amount
- Ensures BBAM_ID is present; inserts it if missing

- Updates AWST and date fields as required

Important: Manual Banner edits (e.g., changing the offer amount) will be overwritten the next time Program 4 runs, as the data flow for offers is one-directional (BAM → Banner).

C. Skipping a Record

A record is skipped if:

- No amount change is required
 - Validation rules fail
 - Record does not qualify as a term-based award
-

11.7 Transaction Reporting

After processing all term-based awards, Program 4 creates a transaction report with:

- Student CWID
- Student Email
- Student Name
- Aid Year
- Fund Code
- Award Amount
- BAM Code (translated)
- Banner AWST Code
- Operation (Added, Updated, Skipped)
- Operation Status

The report is named:

csas_bbam2Banner_term-<date>.txt

Our process copies the reports are automatically moved to the MS-Teams CSAS Scholarships folder.

11.8 Execution & Scheduling

- Script resides on the UC4-accessible execution server
- Scheduled to run immediately after Program 2
- Uses the same infrastructure as Programs 1–3

12. Program 5 — Extract Banner Award Decisions for Return to Blackbaud Awards Management

Program 5 queries Banner to identify any scholarship awards originally imported from BAM that have since been updated in Banner to one of the following statuses:

- **Web Accepted (WA)**
- **Web Declined (WD)**
- **Canceled (C)**

The program extracts only those awards whose decisions have changed within the last **X days** (institution default: **7 days**) and prepares two output files:

1. **A human-readable report** (csas_Banner2bbam-<date>.txt)
2. **A machine-readable CSV** (csas_Banner2bbam.csv) used by the BAM Fund Disbursements Updater API

The script identifies BAM-originated awards through the presence of the BBAM_<id> tag stored in the Banner field FAISMGR.RPRAWRD.RPRAWRD_DATA_ORIGIN.

12.1 Banner Query Logic

The core SQL used to extract updated decision records is based on the following fields from FAISMGR.RPRAWRD:

- RPRAWRD_DATA_ORIGIN
- RPRAWRD_AWST_CODE
- RPRAWRD_AIDY_CODE
- RPRAWRD_ACCEPT_AMT
- RPRAWRD_DECLINE_AMT
- RPRAWRD_CANCEL_AMT
- RPRAWRD_PIDM
- RPRAWRD_ACTIVITY_DATE

The system retrieves rows meeting all of the following conditions:

- RPRAWRD_DATA_ORIGIN LIKE 'BBAM_%'
- RPRAWRD_AWST_CODE IN ('WA','WD','C')
- RPRAWRD_AIDY_CODE = <selected aid year>
- RPRAWRD_ACTIVITY_DATE >= TRUNC(SYSDATE) - <NumberOfDays>

These criteria ensure that:

- Only awards originating from BAM are included
- Only recent decision updates are returned
- Only recognized acceptance/decline/cancel codes are extracted

12.2 Output Files Created by Program 5

Program 5 generates **two output files**, both written to a server directory accessible by Program 6 (the BAM import routine).

A. Text-Based Report File

Filename:

csas_Banner2bbam-<date>.txt

Contents include:

- Student ID (SPRIDEN_ID)
- Email
- Name
- Aid Year
- Amount
- Banner Decision Code
- Date Changed

This file is intended for human review, audit, or troubleshooting.

B. BAM Import File (CSV)

Filename:

csas_Banner2bbam.csv

This is the structured file used by BAM's **Fund Disbursements Updater API**.

It contains the following fields:

Field Name	Description	Source Data
id	BAM Opportunity ID	Parsed from RPRAWRD_DATA_ORIGIN (BBAM_<id><term> pattern)
category_id	Always null	Per institution implementation
initial_category_id	BAM category code mapped from Banner decisions (e.g., Canceled → 77)	Derived from Banner AWST code
award_period_id	Award/Term ID	Parsed from RPRAWRD_DATA_ORIGIN
amount	Amount returned to BAM	RPRAWRD_ACCEPT_AMT for Accepted; null for Declined or Canceled

The integration converts Banner's AWST codes into BAM category IDs according to internal mapping rules (e.g., Canceled → 77).

12.3 Decision → BAM Mapping Logic

The CSV generation includes a translation step:

- **If Accepted:**
 - amount = RPRAWRD_ACCEPT_AMT
 - initial_category_id mapped to BAM's Accepted ID

- **If Declined or Canceled:**
 - amount = null
 - initial_category_id maps to the corresponding BAM Declined or Canceled category ID (e.g., Canceled → 77)

12.4 Summary of Program 5 Responsibilities

Action	Description
Identify BAM-originated Banner awards	Uses RPRAWRD_DATA_ORIGIN LIKE 'BBAM_ %'
Filter by decision changes	Only WA, WD, C, and only within X days
Produce a human-readable report	Includes students, amount, decision, timestamp
Produce machine-readable CSV	Used by BAM Updater API
Prepare files for Program 6	Files are placed in a directory picked up by the next process

13. Program 6 — Import Banner Award Decisions Back into BAM

Program 6 uses the **Fund Disbursements Updater API** to transmit updated Banner award decisions (exported by Program 5) back into Blackbaud Award Management (BAM). The program processes the machine-readable CSV file:

csas_Banner2bbam.csv

and imports its contents into BAM, updating corresponding award records. This is the final step in the BAM ← Banner ← BAM decision-synchronization loop.

13.1 API Execution

The Updater API is invoked using a cURL POST request that uploads the CSV file created in Program 5.

Your source provides the exact call structure:

```
curl --verbose --form "file=@\\NETWORK_DRIVE\\csas_Banner2bbam.csv" \  
https://bigstate.academicworks.com/api/v0/imports/disbursements?token=tokenkey
```

This API call:

- Submits the CSV file containing Banner-generated decision updates
 - Authenticates using the BAM import token
 - Initiates BAM processing of the decisions (Accepted, Declined, Canceled)
-

13.2 Error Handling & Operational Behavior

Per your documentation:

- The interface may **succeed or fail**
- **If it fails, BAM does not provide error detail** through the API response
- You must log in to the BAM administrative portal to determine the failure reason
- Troubleshooting typically requires assistance from **Blackbaud Support**

This is an important operational constraint, as the API does not return granular diagnostics.

13.3 Scheduling & Execution Environment

Program 6:

- Runs on the institution's **cron server**, the same environment used for SFTP processes
- Is scheduled at whatever interval the institution requires (typically daily)

This execution environment must have:

- HTTPS outbound access to the institution's BAM domain
- Permissions to read the CSV dropped by Program 5
- Access to any related logs or report directories
- Network configurations allowing secure API communication

14. Program 7 — Report Distribution to MS Teams

Program 7 is responsible for copying all generated interface reports into the institution's Microsoft Teams channel storage for financial aid reconciliation and recordkeeping.

Specifically, the following reports are transferred:

- csas_Banner2bbam-<date>.txt
- csas_bbam2Banner-<date>.txt

Our reports are placed in:

MS Teams → CSAS → BBAM2CSAS Interface → Files → Interface Reports

This step ensures centralized access by administrators and supports downstream auditing and compliance workflows.

15. Other Institutional Processes

Your environment also includes an additional UC4-scheduled process:

15.1 SZR_ACADEMICWORKS_FILE

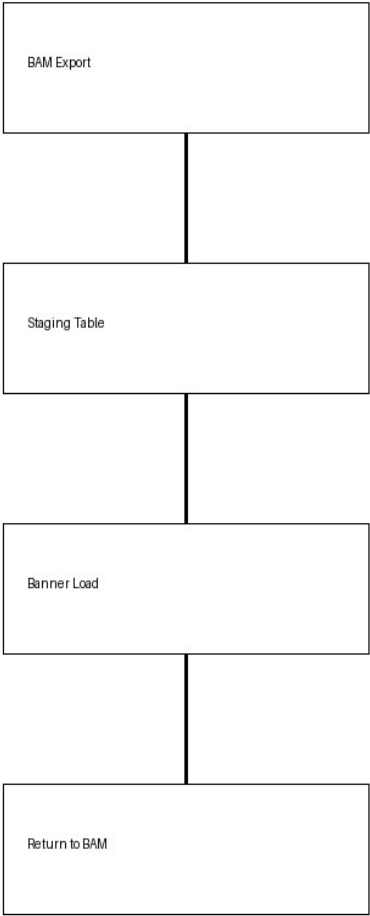
This script:

- Generates a **biodemographic information file** consumed by BAM
- Accepts a **term parameter** to define which student population is included
- Is run **monthly in the spring** to ensure upcoming summer students are added
- May require **manual uploads** of supplemental populations to BAM

This process is outside the primary BAM ↔ Banner award interface but remains part of your institution's total scholarship-system integration workflow.

BAM ↔ Banner Integration – Full Technical Document

This document includes full technical details, flowcharts, summary, conclusion, and glossary.



Summary

The BAM-to-Banner integration automates scholarship award exchange between systems ensuring consistent and accurate financial aid data.

Conclusion

This integration reduces manual work, ensures consistency, and provides a reliable automated pathway for scholarship data synchronization.

Glossary of Terms

BAM (**Blackbaud Award Management**): A scholarship management system used to manage applications, awards, and student interactions.

Banner (**Ellucian Banner ERP**): A student information and financial aid management system used by higher education institutions.

API (**Application Programming Interface**): A system that allows software applications to communicate and exchange data.

UC4 / Automic: An enterprise job scheduler used to automate batch processes.

Fund Disbursement API: BAM API used to export and update award offer and decision data.

RPRAWRD: Banner table used to store annual financial aid award data.

RPRATRM: Banner table used to store term-based financial aid awards.

Staging Table (**COFC_CSAS_AWARDS**): A custom Banner table used to temporarily store BAM data before processing.

Aid Year: A financial aid cycle representing an academic year (e.g., 2023–2024).

BBAM_ID: A unique identifier linking Banner aw

16. Contact Information

Victor C. Swindell

Programmer/Analyst III

Division of Information Technology

College of Charleston

Charleston, SC 29424-0001

Email: Swindellvc@charleston.edu

Available to chat via MS-Teams