A high-level overview of fields synced between PowerCampus and Slate by PowerSlate integration. For exact field names, see *ps\_models.py*.

**Column Key**

* Field Name: Field name in plain English.
* Mapping Source: If translations are involved, where are they stored?
* API/SQL: Is this operation being handled by PowerCampus Web API functionality or custom stored procedures (routines)?

Contents

[Fields Passed Once from Slate -> PowerCampus on Initial Sync 1](#_Toc98947936)

[Fields Updated from Slate -> PowerCampus for Each Sync 2](#_Toc98947937)

[Fields Passed from PowerCampus -> Slate 3](#_Toc98947938)

[Financial Aid Checklist from PowerFAIDS -> Slate 3](#_Toc98947939)

[Tabs Mapped in PowerCampus Mapper 4](#_Toc98947940)

# Fields Passed Once from Slate -> PowerCampus on Initial Sync

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Note* | *Mapping Source* | *API/SQL* |
| Prefix | Does not overwrite existing in PC. | XML | API |
| First Name | Does not overwrite existing in PC. |  | API |
| Middle Name | Does not overwrite existing in PC. |  | API |
| Last Name | Does not overwrite existing in PC. |  | API |
| Suffix | Does not overwrite existing in PC. | XML | API |
| Email | Adds new primary Personal email in PC unless address already exists. |  | API |
| Phone (up to 3 numbers) | Adds new primary phones in PC. Merges with existing numbers of same type. |  | API |
| Address | Adds new primary Home address in PC. |  | API |
| Government ID/TIN/SSN |  |  | API |
| Visa |  | XML | API |
| Primary Citizenship |  | XML | API |
| Secondary Citizenship |  | XML | API |
| Country of Birth |  | XML | API |
| IPEDS Federal Category | This is made up of several fields in both Slate and PC. | Python | API |
| Sex |  | Python | API |
| Marital Status |  | XML | API |
| Veteran/Veteran Status | API 8.8.3 seems broken; passing in 1 will write 2 into [Application].[VeteranStatus]. SQL workaround in place. | XML | API |
| PrimaryLanguage | Language tab in PowerCampus Mapper. | XML | SQL |
| DOB |  |  | API |
| Campus |  | XML | API |
| Program/Degree/Curriculum |  | XML | API |
| Academic Year/Term |  | XML | API |
| ProposedDecision (1:2 mapping to AppStatus + App Decision) | Recommended to set a default (Incomplete/Pending) and use AppStatus and AppDecision instead. Avoids complex mappings in XML. | XML | API |
| Application Submission Timestamp | CreateDateTime in API. Unsure exactly what this does. |  | API |

# Fields Updated from Slate -> PowerCampus for Each Sync

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Note* | *Mapping Source* | *API/SQL* |
| Application Date |  |  | SQL |
| App Status |  | Slate | SQL |
| App Status Date | If not supplied, current date is used. |  | SQL |
| App Decision |  | Slate | SQL |
| App Decision Date | If not supplied, current date is used. |  | SQL |
| Counselor | PEOPLE\_CODE\_ID |  | SQL |
| Department |  | Slate | SQL |
| Nontraditional |  | Slate | SQL |
| Population |  | Slate | SQL |
| Admit Date, Year, Term, Session | Only populates if App Decision is an acceptance value. Admit Date sent from Slate. | Slate | SQL |
| Matric, Date, Year, Term, Session | Matric flag sent from Slate. Date is start date from academic calendar. | Slate | SQL |
| College Attend |  | Slate | SQL |
| Extracurricular |  |  | SQL |
| IPEDS Federal Category | This is made up of several fields in both Slate and PC. | Routine | SQL |
| Ethnicity (Demographics) |  | Slate | SQL |
| Primary Citizenship |  | XML | SQL |
| Secondary Citizenship |  | XML | SQL |
| Marital Status |  | XML | SQL |
| Sex |  | Python | API |
| Visa |  | XML | SQL |
| Veteran |  | XML | SQL |
| PrimaryLanguage | Language tab in PowerCampus Mapper. | XML | SQL |
| HomeLanguage | Language tab in PowerCampus Mapper. | XML | SQL |
| Government ID | Will throw error if different value already exists in PowerCampus. |  | SQL |
| Checklist/Scheduled Actions | Uses stand-alone query in Slate with stuffed, comma-separated application GUID’s as input parameter. | Slate | SQL |
| SMS Opt-In | Admissions Dept opt-in on Telecommunications tab is updated. | Routine | SQL |
| Education Row | Slate export should be nested JSON. A Slate source format can mark School records that were not matched with an org in PowerCampus. Will insert or update; matched by org, degree, and curriculum. Degree and curriculum default to blank. | Routine | SQL |
| Test Scores (Numeric) | Slate export should be nested JSON. Will insert or update; matched by test, test type, and test year/month only.  Total Score in Slate should be named Score0 in export.  Set of numeric scores can be accompanied by single alpha score. Alpha score will be stored with whichever numeric score has matching test type. | Routine | SQL |
| Test Scores (Alpha) | Not yet implemented in Python, although SQL is ready. | Routine | SQL |
| Stops | Slate export should be nested JSON. Will insert or update; matched by stop code and stop date. | Routine | SQL |
| Notes | Inserts a note if a matching note doesn't already exist. | Routine | SQL |
| User-Defined Fields | Inserts or updates data in User Defined fields. | Routine | SQL |

# Fields Passed from PowerCampus -> Slate

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Note* | *Mapping Source* | *API/SQL* |
| PEOPLE\_CODE\_ID | Only if application has been processed through Handle Applications automatically or manually. |  | SQL |
| Record Found | Boolean. Whether or not a matching ACADEMIC record was located. |  | SQL |
| Registered | Boolean. Needed for zero-credit or CEU courses. |  | SQL |
| Registered Credits | From rollup record. |  | SQL |
| Registered Date | PREREG\_VAL\_DATE or REG\_VAL\_DATE | Routine | SQL |
| Readmit | Boolean. True if COLLEGE\_ATTEND matches pc\_readmit\_code defined in config. |  | SQL |
| Withdrawn | Derived from ENROLL\_SEPARATION. | Routine | SQL |
| Campus Email | Configurable by email type code. |  | SQL |

If you want PowerSlate to send changed values to Slate, each field defined in the Active source format needs to be in the Slate applications query with the *compare\_* prefix. For example, *reg\_date* and *compare\_reg\_date*. Python code will compare the values and only send changed values to Slate. If you do not return *compare\_* fields from Slate, all fields for all records will be uploaded to Slate each sync.

# Financial Aid Checklist from PowerFAIDS -> Slate

All data retrieved from PowerFAIDS is retrieved via SQL. Typically, the stored procedures reside in the Campus6 database and pull from the PowerFAIDS database on the same SQL server or on a linked server.

Students are looked up in PowerFAIDS via PCID and SSN (government ID).

If these features are enabled in config and application has been processed through Handle Applications automatically or manually, the following items will be sent to Slate:

|  |  |
| --- | --- |
| *Field/Data* | *Note* |
| Financial Aid Status | Tracking Status. Recommend setting up list of 20 prompts in Slate to match the possible status codes. |
| Financial Aid Checklist | The list of Documents for the award year and their status.  The upload to Slate is a pre-defined, tab-delimited format: <https://technolutions.zendesk.com/hc/en-us/articles/216175018>. Only applications in active Slate rounds will have their FA checklist updated. |
| Financial Aid Awards | An XML representation of Awards. This data is stored in a single Slate text field and intended for display via Liquid looping (not for querying). Sample dashboard briefcase provided.  Awards with net amounts will be shown alongside the full amounts. Awards are broken out into Spring, Summer, and Fall terms by default. Institutions may need to modify the SQL routine to match their terms. |

# Tabs Mapped in PowerCampus Mapper

To achieve the above field integrations, please make sure the following tabs are mapped in recruiterMapping.xml (via the PowerCampus Mapper tool).

Mappings that *might* be optional are italic.

* Country
* State
* AcademicTerm
* AcademicLevel
* AcademicProgram
* Prefix
* Suffix
* *MaritalStatus*
* *Visa*
* *RelationshipType*
* *Campus*
* AddressType
* *Veteran*
* PersonPhoneType
* *FullPartStatus*
* CitizenshipStatus
* ProposedDecision