SIS Baseline Profile (SBP):

The following is a list of the scenarios that comprise an Australian SBP. This profile is currently under development and when formalised will be available at: specification.sifassociation.org/implementation/au/1.2/files/SBP1.0.pdf

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario** | **AU Object element/behavior** | **Message** | **Behavior** | **SBP Rules** |
| 3.1.1 Linking School-centric SIS to existing jurisdiction systems | **SchoolInfo** | **Respond**  | **Respond**  | **A SIS must be able to respond to an unbounded SchoolInfo request.  It should include information about currently open schools and campuses only.** |
| 3.1.2. Future Enrollment | StudentPersonal  | Event  | Create  | Event to be published by SIS.  It is recommended (but not required) that the event should be published in near real-time |
| StudentSchoolEnrollment  | Event  | Create  | SchoolYear: As this is a ‘Future’ enrolment, this value may be greater than the current SchoolYearTimeFrame: =’F’ for future |
| 3.1.3. Future->Current Enrollment  | StudentSchoolEnrollment  | Event  | Change  | Change value of TimeFrame to ‘C’ for CurrentThe TimeFrame value changes when a student becomes current. |
| 3.1.4. New Current Enrollment | StudentPersonal  | Event  | Create  | StudentPersonal needs to be created. |
| StudentSchoolEnrollment  | Event  | Create  | SchoolYear: As this is a “Current” enrolment, this value must be equal to the current SchoolYearTimeFrame: =’C’ for CurrentEntryDate: As this enrolment is current the EntryDate must be the same or less than the day of entry of the information into the SIS |
| 3.1.5. Year End Rollover | StudentSchoolEnrollment | Event | Change | TimeFrame: = ‘H’ for HistoricalRecordClosureReason: = EndOfYear |
|  | StudentSchoolEnrollment | Event | Create | TimeFrame: =’C’ for CurrentSchoolYear: As this is a “Current” enrolment, this value must be equal to the current SchoolYear |
| 3.1.6. Exit School | StudentSchoolEnrollment | Event | Change | TimeFrame: = ‘H’ for Historical |
| 3.1.7. Change Campus | StudentSchoolEnrollment  | Event  | Change (for existing object)  | RecordClosureReason: = CampusExitExitType: Tramsfer campus of the same schoolTimeFrame=’H’ for Historical |
| StudentSchoolEnrollment | Event  | Create (for new object)  | TimeFrame: =’C’ for CurrentSchoolYear: As this is a “Current” enrolment, this value must be equal to the current SchoolYear.EntryType: Transfer from a different campus of the same school |
| 3.1.8. Mid-year Promotion/Demotion | StudentSchoolEnrollment | Event | Change | YearLevel: - Change with new levelPromotionInfo/PromotionStatus: Should be updated with value from value list. |
| 3.2.1. New Staff Member and School Assignment(s) | StaffPersonal  | Event  | Create  | EmploymentStatus: Set to ‘Active’ |
| StaffAssignment  | Event  | Create  | PrimaryAssignment: Set to ‘Y’ for the primary staff assignment |
| 3.2.2 Additional School Assignment (existing Staff Member) | StaffAssignment (pre-existing) | Event | Change  | PrimaryAssignment: Modify where required to reflect status of the new assignment. |
| StaffAssignment (new) | Event  | Create  | PrimaryAssignment: Set to ‘Y’ for the primary staff assignment |
| 3.2.3 Change in Role (e.g. Promotion) | StaffAssignment (pre-existing) | Event | Change  | StaffActivity/Code – UpdatePrimaryAssignment: JobEndDate: |
| StaffAssignment (new) | Event  | Create  | PrimaryAssignment: Set to ‘Y’ for the primary staff assignment |
| 3.2.4 Staff Member leaves School | StaffAssignment  | Event | Change – to end a role  | JobEndDate: is populated for each active StaffAssignment in the school but cannot be pre-dated.Where the school is the organization go to scenario: 3.2.6 |
| 3.2.5 Staff Member goes on leave (extended) | StaffPersonal  | Event  | Change  | EmploymentStatus: Set to “on leave”. No current StaffAssignments should exist but if they do they should be ignored in the zone. |
| 3.2.6 Staff Member leaves Organisation | StaffPersonal  | Event  | Change  | EmploymentStatus: set to "InActive" |
| StaffAssignment  | Event  | Change  | For each StaffAssignment JobEndDate: is populated with Ending Date which cannot be pre-dated. |
| 3.2.7 Change in Personal Details | StaffPersonal  | Event | Change  | Update the changed elements. |
| 3.3.4.1 New Student Contact and Relationship | StudentContactPersonal | Event  | Create  | At least one student whose contact this is must have been previously published. |
| StudentContactRelationship | Event | Create | A new object is published for each student that the Student Contact has a relationship with. |
| 3.3.4.2 New Relationship added to a Current Contact | StudentContactRelationship | Event | Create | Both the Student and the StudentContact must have been previously published. This new object establishes a new relationship between the two entities. |
| 3.3.4.3 Change to a StudentContact Person information | StudentContactPersonal  | Event  | Change  | Update changed elements. |
| 3.3.4.4 Change to StudentContact Relationship Info | StudentContactRelationship  | Event  | Change  | Update changed details including updating ContactFlags where contact priorities have changed. |
| 3.3.4.5 Remove a StudentContact from a Student | **StudentContactRelationship**  | **Event**  | **Delete**  | **This removes the relationship between a Student and a Contact and represents a “true delete” to indicate the severing of the relationship.** |
| 3.4.1. Class-Centric information for publishing by an SIS | TeachingGroup | Respond | Respond | A SIS must be able to respond to a request for all TeachingGroup objects within the zone relating to a particular student or staff member. The StudentPersonal and StaffPersonal objects referenced by the TeachingGroup must already be published (per provisioning scenarios below). If the SIS is reporting on the output of a Timetabling system, the TimeTableCell objects referenced by TeachingGroup must already be published. |
| 3.4.3. Teaching Group set up | TeachingGroup  | Event  | Create  | Either TeachingGroup/StudentList or TeachingGroup/TeacherList contain at least one valid StudentLocalId or StaffLocalId instance TeachingGroup/TeachingGroupPeriodList contains at least one non-empty instance of TeachingGroupPeriod  |
| 3.4.4. Staff Joins Teaching Group | TeachingGroup  | Event  | Change  | TeacherList: Add StaffLocalIdTeachingGroupPeriodList/TeachingGroupPeriod: Remove StaffLocalId to each relevant TeachingGroupPeriodRe-publish lists.Any changes in TeacherList or TeachingGroup PeriodList mean that the Object needs to be published with the complete updated list. |
| 3.4.5. Staff Leaves Teaching Group | TeachingGroup | Event | Change | TeacherList: Remove StaffLocalIdTeachingGroupPeriodList/TeachingGroupPeriod: Remove StaffLocalId to each relevant TeachingGroupPeriodRe-publish lists.Any changes in TeacherList or TeachingGroupPeriodList mean that the Object needs to be published with the complete updated list. |
| 3.4.6. Staff Assignments Change In Teaching Group | TeachingGroup | Event | Change | TeacherList:TeachingingGroupPeriodList:Changes in the Staff member or their Association with a TeachingGroup may change.Re-publish lists.Any changes in TeacherList or TeachingGroupPeriodList mean that the Object needs to be published with the complete updated list. |
| 3.4.7. Student Joins Teaching Group | TeachingGroup | Event | Change | StudentList: Add Student LocalId to list;Re-publish lists.Any changes in StudentList meant that the Object needs to be published with the complete updated list. |
| 3.4.8. Student Leaves Teaching Group | TeachingGroup | Event | Change | StudentList: Remove Student LocalId from list;Re-publish lists.Any changes in StudentList meant that the Object needs to be published with the complete updated list. |
| 3.4.9. Period in Teaching Group Rescheduled | TeachingGroup | Event | Change | TeachingGroupPeriodList: Add/Remove TeachingGroupPeriod from TeachingGroupPeriodList.Re-publish lists.Any changes in TeachingGroupPeriodList mean that the Object needs to be published with the complete updated list. |