



Texas Education Data Standards (TEDS)

Section 9 v2 TSDS Unique ID Specifications

These specifications are valid until March 5, 2018

July 1, 2017

Prepared by: Information Technology Services Business Management Division

Table of Contents

Rollout Plan.....	1
TSDS Unique ID Batch File Format.....	2
Filename	2
Header Record Layout.....	3
Detail Record Layout	4
Trailer Record Layout	7
TSDS Unique ID Web Services Technical Guide.....	8
About This Technical Guide.....	8
About Web Services	8
IDAssignmentV2- Assigning/Retrieving an Identifier	9
Process Flow Example†	9
Operations.....	9
assignId.....	9
Reference WSDL	15
IDAssignmentV2 Transaction Status Table.....	16
NearMatchListV2 - Retrieving Near Match Lists.....	18
Process Flow Example†	18
Operations.....	18
getNearMatches.....	18
Reference WSDL	19
NearMatchListV2 Status Table	20
NearMatchResolutionsV2 – Near Match Decisions	20
Process Flow Example†	20
Operations.....	20
cancel.....	20
assignId.....	21
createNewID	21
Reference WSDL	22
NearMatchResolutionV2 Object Status Table	22
StudentSearch – Searching For a Student / Staff	23
Operations.....	23
searchStudent	23
Reference WSDL	27
Search Status Table	27
BatchInfoSearch – Searching For a Student / Staff.....	29
Operations.....	29
getBatchInformation	29
Batch Statuses.....	29
BatchInfo Search Status Table	30
Appendix A – Soap Message Examples.....	32
IDAssignmentV2 Request – New ID Example.....	32

IDAssignmentV2 Response – New ID Example	33
IDAssignmentV2 Request – Match List Found Example	34
IDAssignmentV2 Response – Match List Found Example	35
NearMatchListV2 Request - Existing Near Match List.....	36
NearMatchListV2 Response – Existing Near Match List	37
NearMatchResolutionV2 Request – Create a New ID	38
NearMatchResolutionV2 Response – Create a New ID.....	38
NearMatchResolutionV2 Request – Assign ID.....	39
NearMatchResolutionV2 Response – Assign ID with Error	39
NearMatchResolutionV2 Request – Cancel.....	40
NearMatchResolutionV2 Response – Cancel	40
Error Response Samples – Invalid Username/Password.....	41
Appendix B – WSDL for Web Services.....	42
IDAssignmentV2 WSDL, operation assigned	44
NearMatchListV2 WSDL, operation getNearMatches	48
NearMatchResolutionV2 WSDL, operations: cancel, assignId, createNewId	51
StudentSearch WSDL, operation search.....	54
Appendix C – IDAssignmentV2 Code Basic Example (VB).....	62

ROLLOUT PLAN

Task	Date
Unique Id v10.1.1 Released without Enrollment Functionality	August 11, 2017
Unique Id v10.2.0 Released without Enrollment Functionality with no impact to v10.1.1 functionality.	September 15, 2017
Unique Id v10.3.0 Released with Enrollment Functionality	March 5, 2018

TSDS UNIQUE ID BATCH FILE FORMAT

Detailed below are the specifications of the TSDS Unique ID Batch File Format. This file format must be followed when submitting students/staff for id assignment (input file). The Batch File should contain three different categories of information and conform to the file naming conventions. The three categories of records are:

- Header Record
- Detail Record
- Trailer Record

The Header and Trailer record should be delimited by a single tab or space character. The Trailer shall have no empty spaces or data after the last Trailer column.

The same file format will be followed when the TSDS Unique ID system generates a file with the TSDS Unique Id number assignment (output file).

Filename

For each Batch file submitted, the system must identify the correct organization and data collection in order to process the file correctly:

Student:

Position	1-6	7	8-10	11	12-18	19	20-31
Description	District Code	_	Campus ID	_	Collection Code	_	Timestamp

Staff:

Position	1-6	7	8-10	11	12-16	17	18-29
Description	District Code	_	Campus ID	_	Collection Code	_	Timestamp

District Code: The LEA code for which the data is being uploaded. (6 digits in length)

Campus ID: The campus ID for which the data is being loaded. If loading data for the LEA, use "000" for campus ID. (3 digits in length).

Collection Code: The collection code is a character string that uniquely identifies the data collection. (8 characters in length)

- The Collection Code shall indicate the type of data being submitted. The values are:
 - “STUDENT” which represents the Student data.
 - “STAFF” which represents the Staff data.

Timestamp: The timestamp is a date in YYYYMMDDHHMM format (e.g. 201406021015). The Timestamp shall be a system generated value at the time the data is extracted. (12 digits in length)

There **must** be an underscore (_) between each element in the file name.

Examples:

227901_002_STUDENT_201201241015 = Student Data for Austin High School in Austin ISD.

227901_000_STAFF_201201241159 = Staff Data for Austin ISD.

Header Record Layout

The header record should always be the first record. Each of the fields should be delimited in the header by the field name reference below. The header record contains the following fields:

Field Name	Delimiter	Required	Data Type	Format Details
Record Type		Yes	VarChar (2)	The Record Type Value must equal "TH".
Extract Date		Yes	VarChar (10)	The Extract Date must be in one of the following formats: <ul style="list-style-type: none"> • mm/dd/yyyy • m/d/yyyy • mm/d/yyyy • m/dd/yyyy • yyyy-mm-dd
Extract Time		Yes	VarChar (8)	The Extract Time must be in the following format: <ul style="list-style-type: none"> • hh:mm:ss
Transmission ID		Yes	VarChar (10)	The Transmission ID is a number utilized by the submitter for auditing as an identifier for the submitter.
Version		Yes	VarChar (10)	The Version Value must equal "2.1".
Delimiter	delimiter=0X2C	Yes	VarChar (4)	The Delimiter must be a comma delimiter, the character literal would be "delimiter=," and the hex representation would be "delimiter=0X2C".
Source Name	source=	Yes		The Source Name value must equal "SIS" or "HR". If a source is not provided in the header, the file will be rejected.
Text Qualifier	qualifier=0X22	Yes		The Text Qualifier must be the text or the hex representation of the text qualifier that is used in the detail records (double quote only qualifier=0X22)

Detail Record Layout

The detail records appear between the header and trailer records. Each of the fields should be delimited even if data is not available for certain fields. These records should represent the individual student/staff data and should contain the following fields:

Field Name	Required	Data Type	Used for Matching	Format Details
Record Type	Yes	VarChar (2)	N/A	The Record Type Value must equal "ID".
Current School Code	Yes	VarChar (9)	No	School Code where the student is currently enrolled or where the staff is currently employed. Indicates the unique campus number registered with the Texas Education Directory (askTED). Note: For non-campus staff, submit 000 as the last three digits of the Current School Code (for example, 227901000).
Reserved Field	No	VarChar (20)	No	Not used by TEA. Leave this field blank.
Last Name	Yes	VarChar (25)	Yes	Legal last name of the student or staff.
First Name	Yes	VarChar (17)	Yes	Legal first name of the student or staff.
Middle Name	No	VarChar (14)	Yes	Legal middle name of the student or staff.
Name Suffix	No	VarChar (1)	No	Valid Values: 1 (Jr), 2 (Sr), 3 (II), 4 (III), 5 (IV), 6 (V), 7 (VI), 8 (VII), 9 (VIII), A (I), B (IX), C (X)
Gender	Yes	VarChar (1)	Yes	Valid Values: F (Female) or M (Male).
Date of Birth	Yes	VarChar (10)	Yes	Valid Formats: <ul style="list-style-type: none"> • mm/dd/yyyy • m/d/yyyy • mm/d/yyyy • m/dd/yyyy • yyyy-mm-dd
Current Grade Level	Yes	VarChar (2)	No	Valid Values: EE (Early Education), PK (Pre-Kindergarten), KG (Kindergarten), 01 (Grade 1), 02 (Grade 2), 03 (Grade 3), 04 (Grade 4), 05 (Grade 5), 06 (Grade 6), 07 (Grade 7), 08 (Grade 8), 09 (Grade 9), 10 (Grade 10), 11 (Grade 11), 12 (Grade 12), and 00 (only valid for Staff)
Local Student ID	Yes	VarChar (9)	No	ID used in the local Student Information System to uniquely identify the student in the submitting system. The primary purpose of this field is to provide a mechanism to import student data from the TSDS Unique ID system back into the local source systems. For staff, ID used in the local Human Resource System.
Social Security Number or S-#	Yes	VarChar (9)	Yes	Student or Staff Social Security Number. Or, state-approved alternate identification for student.
Race	Yes	VarChar (1)	No	Valid Values: I (American Indian Alaska Native), A (Asian), B (Black African American), P (Native Hawaiian Pacific Islander), W (White).
TSDS Unique ID	No	VarChar (10)	No	This field is the assigned unique id and is provided in the output file. This value will be ten significant digits and will not begin with 0.

Current District Yes VarChar (6) No District code where the student is currently enrolled

Field Name	Required	Data Type	Used for Matching	Format Details
Code				or where staff is currently employed. Indicates the district identification number registered with the Texas Education Agency.
Current School Year	Yes	VarChar (4)	No	Valid Format: YYYY. Represents the latest year in the school year range. ("2013" for the 2012-2013 school year).
Alternate Last Name	No	VarChar (25)	No	Maiden Name of Student or Staff.
Reserved Field	No	VarChar (50)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (60)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (25)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (25)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (25)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (25)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (25)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (25)	No	Not used by TEA. Leave this field blank.
Reserved Field	No	VarChar (25)	No	Not used by TEA. Leave this field blank.
Record Status	No	VarChar (255)	No	This field is an internal transaction status result for the record and is provided in the output file. This field must be blank in the input file.
Record Reference Number	No	VarChar (12)	No	This field is an internal transaction identifier for the record and is provided in the output file. This field must be blank in the input file.
Ethnicity Indicator	Yes	VarChar (1)	No	Valid Values: 1 or 0, 1 indicates a person of Cuban, Mexican, Puerto Rican, South or Central American, or other Spanish culture or origin, regardless of race. 0 indicates a person not Hispanic/Latino.
Race 2 Code	No	VarChar (1)	No	Valid Values: I (American Indian Alaska Native), A (Asian), B (Black African American), P (Native Hawaiian Pacific Islander), W (White). See Example.
Race 3 Code	No	VarChar (1)	No	Valid Values: I (American Indian Alaska Native), A (Asian), B (Black African American), P (Native Hawaiian Pacific Islander), W (White). See Example.
Race 4 Code	No	VarChar (1)	No	Valid Values: I (American Indian Alaska Native), A (Asian), B (Black African American), P (Native Hawaiian Pacific Islander), W (White). See Example.
Race 5 Code	No	VarChar (1)	No	Valid Values: I (American Indian Alaska Native), A (Asian), B (Black African American), P (Native Hawaiian Pacific Islander), W (White). See Example.

Record Update Date	No	VarChar (10)	No	This field is the last update for the record and is provided in the output file. This field must be blank in the input file.
-----------------------	----	--------------	----	--

The Cancelled Records File contains all the fields in the Detail Record in addition to the following field:

Record Comments	No	VarChar (2048)	No	This field contains a list of validation errors that the application encountered for this input record during the validation process. If more than one error was found in a record, all errors are listed with each one delimited from the other by a semicolon.
-----------------	----	----------------	----	--

NOTE: One batch file should not contain more than 40,000 detail records.

Trailer Record Layout

The trailer record should always be the last record in the file and should contain the following fields:

Field Name	Required	Data Type	Format Details
Record Type	Yes	VarChar (2)	The Record Type Value must equal "TT".
Transmission ID	Yes	VarChar (10)	The Transmission ID is a number utilized by the submitter for auditing as an identifier for the submitter. This ID must match the Transmission ID in the header record.
Number of Records	Yes	VarChar (10)	Number of records in the file, including the "TH" and "TT" records. The value should be left aligned and not have any trailing spaces.

There should be no extra spaces or data after the last column in the Trailer Record.

Example for Race Codes:

Five Race fields are available in the application, but the same value cannot currently be supplied in any of the five fields. For example, if "A" is supplied in the Race field, "A" cannot be supplied in the Race 2, Race 3, Race 4, or Race 5 fields. Additionally, data must be supplied in consecutive order and a race column cannot be skipped. For example, a race code cannot be supplied in the Race 4 field if there is not a value in the Race, Race 2, or Race 3 fields.

Valid Values: I (American Indian Alaska Native), A (Asian), B (Black African American), P (Native Hawaiian Pacific Islander), W (White).

Example	Race	Race 2	Race 3	Race 4	Race 5
Black	B				
Black, Asian	B	A			
Black, Asian, White	B	A	W		
White, Asian	W	A			
White, Black	W	B			
White, Black, Asian, American Indian, Pacific Islander	W	B	A	I	P

Sample File:

This is a comma delimited, double quote qualified file with a header and a trailer record.

```
TH 10/05/2012 10:42:35 TEAPID 2.1 delimiter=0X2C source=SIS qualifier=0X22
"ID","999999101","","Student","Jane","","","F","3/2/2009","EE","557000","S00005678","W","","999999","2012","","",
"","","","","","","","","","","1","","","",""
TT TEAPID 3
```

TSDS UNIQUE ID WEB SERVICES TECHNICAL GUIDE

About This Technical Guide

This technical guide serves as a reference guide for the TSDS Unique ID Web Services for Students and Staff. A Web Service is designed to allow different applications from different sources to communicate with TSDS Unique ID using an XML-based process. These Web Services allow TSDS Unique ID system functionality to be exposed for assigning or retrieving identifiers, requesting a list of near matches, submitting near match decisions, and searching by student /staff or batch.

This document explains in detail the information necessary to utilize the TSDS Unique ID Web Services (Web Services). In this document, you will learn:

- How the TSDS Unique ID Web Services work.
- How to interpret the response information sent back by the TSDS Unique ID Web Service.

Submissions of students and staff through the Web Service are logged as a separate batch type, thus allowing for filtering within the TSDS Unique ID system interface later.

This technical guide includes some notes that are designed to highlight important information.

The following Notes indicators are used:



Note: *This box mentions things that require special attention. The symbol to the left indicates an important note to remember.*

About Web Services

TSDS Unique ID includes four Web Services, including ID Assignment, Near Match List, Near Match Resolution, Search by Batch, and Search By Student/Staff. The first three Web Services existed in prior releases and will function as they did previously. The system includes two file formats for Version 7.x and above. To accommodate the new template, new Web Services have been created for ID Assignment, Near Match Resolution, and Near Match List. These new Web Services are identified with V2 in the service names and object names. For example, the new ID Assignment Web Service is IDAssignmentV2 and the Student object is StudentV2. The existing Web Services have not changed.

IDAssignmentV2- Assigning/Retrieving an Identifier

The IDAssignmentV2 Web Services will complete the identifier assignment process and assign an identifier, return a list of near matches, update history information for matching records, and will allow a state identifier to be returned. Upon submission of a request, the Web Service will return an ID Assignment Result element. It will be named IDAssignmentResultV2 depending on the service being called. This contains the status property that identifies success values for the request.

If the submission record does not have a matching record in TSDS Unique ID system, the ID Assignment Result element will return a no match status and will assign an ID. The newly assigned ID will be returned in the StateStudentId element.

Example:

```
<stateStudentId xsi:type="xsd:string">5202763107</stateStudentId>
```

If a matching record is found in the system, the history information for the record will be updated if needed as appropriate and will return a status message indicating the history update. If a matching record is found in the system, but the no updates were needed, a status will be returned indicating no update was needed. If the submission record cannot be uniquely identified by the system, the Web Service will return the potential match records for the submission. For more details, please refer to the IDAssignmentResultV2 Status table below and the SOAP Examples in Appendix A.

Web service accounts for LEAs are managed in the TEAL secure environment. These service account credentials are passed in the web service call as parameters.

Process Flow Example†

1. User is working in a system outside of TSDS Unique ID system to submit student data to the state.
2. User gets an error in the other application stating that a student does not have an identifier.
3. User requests an identifier from TSDS Unique ID directly from that other system by using that applications functionality.
4. Other system uses IDAssignmentV2 Web Service to request an identifier from TSDS Unique ID for Students.
5. TSDS Unique ID processes the request, stores the student information in the database, and returns the new or pre-existing identifier to the originating application.

† Indicates that the example provided is only one example of a possible scenario. Other scenarios may exist that could demonstrate the flow process for the web service.

Operations

assignId

The assignId operation of the IDAssignmentV2 services will assign an identifier or return potential matches for a submission, as well as update data as necessary.

Input Elements

- ❖ **STUDENT**-This element will represent the student details for the identifier assignment request. This will be Student or StudentV2 depending on the Web Service used. The student input elements are:

Field/Element	Description	Notes
SSN	Required. <i>Valid Values/Format:</i> Must be 9 digits, can include "-" but must be formatted correctly with three, two and four digits. <i>Example:</i> 444335555 OR 444-33-5555 Or 8 digits with the character "S" in front. Example S44335555	V1 and V2
birthDate	Required. <i>Valid Values/Format:</i> yyyy-mm-dd; MM/DD/YYYY; <i>Example:</i> 2000-10-30	V1 and V2
districtCode	Required. <i>Valid Values/Format:</i> Must be valid based on user authentication and location.	V1 and V2
gender	Required. <i>Valid Values/Format:</i> F or M	V1 and V2
gradeLevel	Required. <i>Valid Values/Format:</i> EE (Early Education), PK (Pre-Kindergarten), KG (Kindergarten), 01 (Grade 1), 02 (Grade 2), 03 (Grade 3), 04 (Grade 4), 05 (Grade 5), 06 (Grade 6), 07 (Grade 7), 08 (Grade 8), 09 (Grade 9), 10 (Grade 10), 11 (Grade 11), 12 (Grade 12), and 00 (only valid for Staff)	V1 and V2
localStudentId	Required. <i>Valid Values/Format:</i> Length must be less than or equal to 20 characters. <i>Example:</i> 4857390022	V1 and V2
raceEthnicCode	Required. . <i>Valid Values/Format:</i> I A B P W	V1 and V2
residentDistrictCode	Optional. <i>Valid Values/Format:</i> Must be valid based on user authentication and location.	V1 and V2
schoolCode	Required. <i>Valid Values/Format:</i> Must be valid based on user authentication and district.	V1 and V2
schoolYear	Required. <i>Valid Values/Format:</i> Must be any 4 digit number, cannot start with 0. <i>Example:</i> 2008	V1 and V2
stateStudentId	Optional. <i>Valid Values/Format:</i> Must be numeric with a length of 10 or 12	V1 and V2

Field/Element	Description	Notes
	characters.	
<i>studentFirstName</i>	Required. <i>Valid Values/Format:</i> Length must be less than or equal to 60 characters.	V1 and V2
<i>studentLastName</i>	Required. <i>Valid Values/Format:</i> Length must be less than or equal to 60 characters.	V1 and V2
<i>studentMiddleName</i>	Optional. <i>Valid Values/Format:</i> Length must be less than or equal to 60 characters.	V1 and V2
<i>studentSuffix</i>	Optional. <i>Valid Values/Format:</i> : 1 (Jr), 2 (Sr), 3 (II), 4 (III), 5 (IV), 6 (V), 7 (VI), 8 (VII), 9 (VIII), A (I), B (IX), C (X)	V1 and V2
<i>alternateID</i>	Optional <i>Valid Values/Format:</i> Must be numeric with a length of 50 characters. Note- This field is required if the alternateSource is submitted.	V2 only
<i>alternateSource</i>	<i>Valid Values/Format:</i> Must be numeric with a length of 60 characters. Note- This field is required if the alternateID is submitted.	V2 only
<i>previousLastName</i>	Optional <i>Valid Values/Format:</i> Length must be less than or equal to 60 characters.	V2 only
<i>sourceSISCode</i>	Required <i>Valid Values/Format:</i> HR or SIS	V2 only
<i>cdfValuesArray</i>	Optional <i>Valid Values/Format:</i> Length must be less than or equal to 25 characters. Note- CDFs are not overwritten if subsequent submissions are blank. To overwrite CDFs NULL must be sent as the value.	V2 only
<i>cdfNumberArray</i>	Optional <i>Valid Values/Format:</i> Length must be less than or equal to 25 characters. To be used to identify the CDF number associated to the <i>cdfValuesArray</i> values passed. For example, if 3 CDF values are being passed, the <i>cdfNumberArray</i> should indicate which value should be assigned to the specific CDF numbers. CDFs range from 1 to 8.	V2 only
<i>ethnicityIndicator</i>	Required <i>Valid Values/Format:</i> 0 or 1	V2 Only
<i>Race2Code</i>	Optional <i>Valid Values/Format:</i> I A B P W	V2 Only
<i>Race3Code</i>	Optional <i>Valid Values/Format:</i> I A B P W	V2 Only

Field/Element	Description	Notes
<i>Race4Code</i>	Optional <i>Valid Values/Format: I A B P W</i>	V2 Only
<i>Race5Code</i>	Optional <i>Valid Values/Format: I A B P W</i>	V2 Only

- ❖ **USER ID - Required.** This value will be a string that will identify the user ID that performed the action. Must be a Super User account.
- ❖ **USER PASSWORD - Required.** This value will be the user ID's password that performed the action. Must be a Super User account.

Output Elements

Upon completion of the request, the *IDAssignmentResultV2 element* is returned. This will be IDAssignmentResultV2 depending on the service called. The IDAssignmentResultV2 output will have the following elements in it:

- ❖ *Status* – this is a string containing the status of the function call. For a full list of status values refer to the IDAssignmentV2 Status Table below.
- ❖ *StateStudentId* – use this property to get the state id assigned to the student.
- ❖ *SubmissionId* – use this property to get the submission id assigned to the transaction.
- ❖ *BatchNumber* – use this property to get the batch number assigned to the transaction.
- ❖ *cdfMap* – used internally for CDFs.
- ❖ *aliasId* – this property returns the alias ID for the record.
- ❖ *PotentialNearMatches* – this property will return an array of PotentialMatch elements. The data that is returned in this array is for the near matching (master) records. Each near match element will have the following properties:
 - a. STATE STUDENT ID (stateStudentID) – string with the state student Identifier of the matched record.
 - b. LAST NAME (lastName) – the last name of the matched record.
 - c. FIRST NAME (firstName) – the first name of the matched record.
 - d. MIDDLE NAME (middleName) – the middle name of the matched record.
 - e. SUFFIX (suffixName) – the suffix of the matched record.
 - f. GENDER (gender) – the gender code of the matched record.
 - g. DATE OF BIRTH (dob) – the date of birth of the matched record.
 - h. ETHNIC CODE (ethnicCode) – the ethnic code of the matched record.
 - i. SOCIAL SECURITY NUMBER (ssn) – the social security number of the matched record.
 - j. MATCH PROBABILITY (matchProbability) – the match score ranges from Lower Near Match Threshold to Upper Near Match Threshold setting from System Properties.
 - k. COMMENTS (comments) – the comments for the matched record.
 - l. CREATION DATE (createDate) – the date the record was created.
 - m. CLAIM REQUIRED (claimRequired) – indicates if a claim is required for the record. This can occur when Claiming is enabled.
 - n. DISTRICT CODE (districtCode) – the District Code of the matched record.
 - o. SCHOOL CODE (schoolCode) – the School Code of the matched record.
 - p. SCHOOL YEAR (schoolYear) – the school year of the matched record.
 - q. GRADE LEVEL (gradeLevel) – the grade level of the matched record.
 - r. CDF NUMBERS ARRAY (cdfNumbersArray) – the array of CDF by number (V2 only)
 - s. CDF VALUES ARRAY (cdfValuesArray) – the array of CDF values (V2 only)
 - t. CDF REF NUMBER (cdfRefNumber) (V2 only) – the internal reference to the CDF.
 - u. ALIAS ID (aliasID) (V2 only) – the alias ID of the matched record.
 - v. ALTERNATE SOURCE (alternateSource) (V2 only) – the alternate source of the matched record.
 - w. PREVIOUS LAST NAME (previousLastName) (V2 only) – the previous last name of the matched record.

-
- x. SOURCE SIS CODE (sourceSISCode) (V2 only) – the source system of the matched record.
 - y. ETHNICITY INDICATOR (ethnicityIndicator) (V2 only) – the ethnic code for the matched record.
 - z. RACE 2 CODE (Race2Code) (V2 only) – the race 2 code value for the matched record.
 - aa. RACE 2 CODE Description (Race2CodeDesc) (V2 only) – the race 2 code value description for the matched record.
 - bb. RACE 3 CODE (Race3Code) (V2 only) – the race 3 code value for the matched record.
 - cc. RACE 3 CODE Description (Race3CodeDesc) (V2 only) – the race 3 code value description for the matched record.
 - dd. RACE 4 CODE (Race4Code) (V2 only) – the race 4 code value for the matched record.
 - ee. RACE 4 CODE Description (Race4CodeDesc) (V2 only) – the race 4 code value description for the matched record.
 - ff. RACE 5 CODE (Race5Code) (V2 only) – the race 5 code value for the matched record.
 - gg. RACE 5 CODE DESCRIPTION (Race2CodeDesc) (V2 only) – the race 5 code value description for the matched record
- ❖ *fieldsWithErrors* – returns an array of field errors. The possible messages are:
- a. LAST_NAME – the submitted value was too long or null.
 - b. FIRST_NAME – the submitted value was too long or null.
 - c. MIDDLE_NAME – the submitted value was too long.
 - d. SUFFIX – the submitted value was too long.
 - e. GENDER – the submitted value was too long, invalid, or null.
 - f. DATE_OF_BIRTH – the submitted value was too long, invalid, or null.
 - g. ETHNICITY – the submitted value was too long or invalid.
 - h. SSN – the submitted value was invalid.
 - i. GRADE_LEVEL - the submitted value was too long, invalid, or null.
 - j. LOCAL_STUDENT_ID – the submitted value was too long, invalid, or null.
 - k. SCHOOL_YEAR – the submitted value was too long, invalid, or null.
 - l. DISTRICT_CODE – the submitted value was too long, invalid, or null.
 - m. SCHOOL_CODE – the submitted value was too long, invalid, or null.
 - n. RES_DISTRICT_CODE – the submitted value was too long, invalid, or null.
 - o. STATE_STUDENT_ID – the submitted value was too long or invalid.
 - p. SOURCE_SYSTEM – the submitted value was too long or invalid. (V2only)
 - q. ALTERNATE_SOURCE – the submitted value was too long or invalid. (V2 only)
 - r. ALTERNATE_ID – the submitted value was too long or invalid. (V2 only)
 - s. ALTERNATE_LAST_NAME - the submitted value was too long. (V2 only)
 - t. CDF_XXXXX – the submitted value was too long (V2 only). XXXXX stands for the name of the CDF field.
 - u. AGENCY_NOT_FOUND – the submitted value was invalid.
 - v. AGENCY_NOT_AUTHORIZED – the user is not authorized to access this agency.
-

Reference WSDL

<https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2?WSDL>

Shown in Appendix B

Example: <https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2?WSDL>

IDAssignmentV2 Transaction Status Table

Status Code/Message	Description	Notes
01	Ready to Validate Data	
02	Ready to Assign IDs	
03	Ready to Fix Data Errors	
04	Canceled During Fix Errors Stage	
05	If this value is returned then the student identifier can be retrieved via the StateStudentId operation. New ID Assigned - No Matching Record Found	
06	Ready to Resolve Near Matches/Duplicates If this value is returned the system was not able to assign an identifier. The Web Service will return the list of the potential matches that can be retrieved via the PotentialNearMatches operation.	
07	New ID Assigned During Match Resolution Stage	
08	Student Found and History Created During Match Resolution Stage	
09	Canceled During Match Resolution Stage	
10	No Information Available -ID Retired	
11	Student Found and History Created -Exact Match If this value is returned the system found a match and updated the system. The state student identifier can be retrieved via the StateStudentId operation.	
12	Student Found but No Change in Data -Exact Match If this value is returned the system found a match but there were no updates needed in the system. The state student identifier can be retrieved via the StateStudentId operation.	
13	Canceled During Data Validation Stage -ID Contained in Record Not Found in System	
14	Ready to Assign IDs -Exact Match	
15	Waiting for Claim Response -Auto Claim Created	
16	ID Updated and History Created -Auto Claim Created and Accepted	
17	ID Not Updated -Auto Claim Created but Disputed	
18	Canceled During Assign ID Stage -Claim Required	
19	Canceled During Data Validation Stage	
20	Canceled During Assign ID Stage	
21	Ready to Assign IDs -ID Present in Record -No Exact Match	
22	Ready to Assign IDs -Possible Claim	
23	Canceled During Assign ID Stage -Claim Already Exists	
24	ID Not Updated -Auto Claim Created but Deleted	
25	Canceled During Claim Creation	
26	Student Found During Assign ID Stage but History Not Created	

Status Code/Message	Description	Notes
27	Student Found During Match Resolution Stage but History Not Created	
28	New ID Assigned -Administrator	
29	Ready to Validate Data -Search	
30	Ready to Search IDs	
31	Canceled During Fix Errors Stage -Search	
32	No Matching Record Found -Search	
33	Canceled During ID Search Stage -Multiple Records Found	
34	Matching Record Found -Search	
35	Canceled During Data Validation Stage -Search	
36	Canceled During ID Search Stage -Near Match(es) Found	
37	Ready to Fix Data Errors -Search	
38	Ready to Assign IDs -Possible Claim Found During Assign ID Stag	
39	ID Updated -Administrator	
40	Student Found and History Created During Assign ID Stage	
41	Student Updated and History Created -Direct Edit	
42	Canceled During Data Validation Stage -Claim Required	
43	Existing ID Assigned During Match Resolution Stage -No Exact Match	V2 ONLY
44	New ID Created During Match Resolution Stage -No Exact Match	V2 ONLY
FAILED_USER_NOT_FOUND_OR_INACTIVE	The call failed because the user id passed was not found in the system or the user id is not active.	
FAILED_VALIDATION	One or more fields have a validation error. Use the fieldsWithErrors array to get a list of the fields.	
FAILED_USER_NOT_FOUND_ERROR	The call failed because the user id was null.	
LOCKOUT	When the system is locked out due to an Administrator lockout or forced lockout by a Matching Engine issue, the lockout status will be returned.	
FAILED_USER_NOT_AUTHORIZED	The call failed because the user id is not authorized to perform the request.	
FAILED_REQUIRED_CDF_MISsing	The call failed because one or more required customer defined fields were not submitted.	
SUCCESS_NO_MATCHING_RECORDS	The call was successful but no matches were found.	
SUCCESS_MULTIPLE_MATCHING_RECORDS	The call was successful and multiple matches were found.	
SUCCESS_SINGLE_MATCH_RECORD	The call was successful and a single match record was found.	

NearMatchListV2 - Retrieving Near Match Lists

The Near Match List Web Service will return a list of near matches found by the TSDS Unique ID system for an input record submitted.

The Web Service will have one operation called NearMatchListV2. Upon submission of a request, the Web Service will return a NearMatchListResult element. The NearMatchListResult element will have a status and an array of PotentialMatch elements.

Process Flow Example†

1. User is working in a system outside of TSDS Unique ID to submit student data to the state.
2. User gets an error in the other application stating that a student does not have an identifier.
3. User requests an identifier from TSDS Unique ID directly from that other system by using that applications functionality.
4. Other system uses the IDAssignmentV2 Web Service to request an identifier from TSDS Unique ID.
5. TSDS Unique ID processes the request, finds one or more near matches/matches and returns the list of near matches to the other application.
6. User can view a list of near matches in the other system and resolve or can retrieve the list later.

† Indicates that the example provided is only one example of a possible scenario. Other scenarios may exist that could demonstrate the flow process for the web service.

Operations

getNearMatches

This operation will return potential match records that matched the submission record during the IDAssignmentV2 request.

Input Elements

- ❖ **SUBMISSION IDENTIFIER –Required.** This value will be the identifier that was assigned to submission record during IDAssignmentV2.
- ❖ **USER ID - Required.** This value will be a string that will identify the user ID that performed the action. Must be a Super User account.
- ❖ **USER PASSWORD - Required.** This value will be the user ID's password that performed the action. Must be a Super User account password.

Output Elements

Upon completion of the NearMatchListV2 request, the *NearMatchListResult element* is returned. The *NearMatchListResult* element will have the following properties:

- ❖ **Status** – this is a string containing the status of the function call. For a full list of status values refer to the NearMatchListV2 Status Table below.
- ❖ **Array Of PotentialMatchObjects** - Each near match element will have the following properties:
 - a. **STATE STUDENT ID (STATESTUDENTID)** – string with the state student Identifier of the matched record.
 - b. **LAST NAME (LASTNAME)** – the last name of the matched record.
 - c. **FIRST NAME (FIRSTNAME)** – the first name of the matched record.
 - d. **MIDDLE NAME (MIDDLENAME)** – the middle name of the matched record.
 - e. **SUFFIX (SUFFIXNAME)** – the suffix of the matched record.

- f. GENDER (GENDER) – the gender code of the matched record.
- g. DATE OF BIRTH (DOB) – the date of birth of the matched record.
- h. ETHNIC CODE (ETHNICCODE) – the ethnic code of the matched record.
- i. SOCIAL SECURITY NUMBER (SSN) – the social security number of the matched record.
- j. MATCH PROBABILITY (MATCHPROBABILITY) – the match score ranges from Lower Near Match Threshold to Upper Near Match Threshold setting from System Properties.
- k. COMMENTS (COMMENTS) – the comments for the matched record.
- l. CREATION DATE (CREATIONDATE) – the date the record was created.
- m. CLAIM REQUIRED (CLAIMREQUIRED) – indicates if a claim is required for the record. This can occur when Claiming is enabled..
- n. DISTRICT CODE (DISTRICTCODE) – the District Code of the matched record.
- o. SCHOOL CODE (SCHOOLCODE) – the School Code of the matched record.
- p. SCHOOL YEAR (SCHOOLYEAR) – the school year of the matched record.
- q. GRADE LEVEL (GRADELEVEL) – the grade level of the matched record.
- r. CDF NUMBERS ARRAY (cdfNumbersArray) – the array of CDF by number (V2 only)
- s. CDF VALUES ARRAY (cdfValuesArray) – the array of CDF values (V2 only)
- t. CDF REF NUMBER (cdfRefNumber) (V2 only) – the internal reference to the CDF.
- u. ALIAS ID (aliasID) (V2 only) – the alias ID of the matched record.
- v. ALTERNATE SOURCE (alternateSource) (V2 only) – the alternate source of the matched record.
- w. PREVIOUS LAST NAME (previousLastName) (V2 only) – the previous last name of the matched record.
- x. SOURCE SIS CODE (sourceSISCode) (V2 only) – the source system of the matched record.
- y. ETHNICITY INDICATOR (ethnicityIndicator) (V2 only) – the ethnic code for the matched record.
- z. RACE 2 CODE (Race2Code) (V2 only) – the race 2 code value for the matched record.
- aa. RACE 2 CODE Description (Race2CodeDesc) (V2 only) – the race 2 code value description for the matched record.
- bb. RACE 3 CODE (Race3Code) (V2 only) – the race 3 code value for the matched record.
- cc. RACE 3 CODE Description (Race3CodeDesc) (V2 only) – the race 3 code value description for the matched record
- dd. RACE 4 CODE (Race4Code) (V2 only) – the race 4 code value for the matched record.
- ee. RACE 4 CODE Description (Race4CodeDesc) (V2 only) – the race 4 code value description for the matched record.
- ff. RACE 5 CODE (Race5Code) (V2 only) – the race 5 code value for the matched record.
- gg. RACE 5 CODE DESCRIPTION (Race2CodeDesc) (V2 only) – the race 5 code value description for the matched record

Reference WSDL

<https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2?WSDL>

Shown in Appendix B.

Example: <https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2?WSDL>

NearMatchListV2 Status Table

Status Code	Description
ERROR	An exception occurred.
FAILED_USER_NOT_FOUND_OR_INACTIVE	The call failed because the user id passed was not found in the system or the user id is not active.
FAILED_STAGE_RECORD_NOT_PENDING_NEAR_MATCH	The submission record is no longer pending the near match status.
FAILED_STAGE_RECORD_NOT_FOUND	The submission record cannot be found in the database.
SUCCESS	The call returned one or more potential match records.
FAILED_USER_NOT_AUTHORIZED	The call failed because the user id was null.
FAILED_INVALID_SUBMISSION_ID	The submission id has an invalid format.

NearMatchResolutionsV2 – Near Match Decisions

The Near Match Resolution Web Service will accept and process a near match decision. The Web Service will perform one of the following actions based on the decision:

1. Cancel – cancel the assignment request.
2. Assign Pre-Existing Identifier – assign an already existing state student identifier to the submission record.
3. Create New ID – create a new state student identifier for the submission record.

Process Flow Example†

1. User is working in a system outside of TSDS Unique ID to submit student data to the state.
2. User gets an error in the other application stating that a student does not have an identifier.
3. User requests an identifier from TSDS Unique ID directly from that other system by using that applications functionality.
4. Other system uses IDAssignmentV2 Web Service to request an identifier from TSDS Unique ID.
5. TSDS Unique ID processes the request, finds one or more near matches/matches and returns the list of near matches to the other application.
6. User is presented with the list of near matches in the other system and makes a decision.
7. Other system uses web services to submit the near match decision.
8. TSDS Unique ID processes and stores the near match decision.

† Indicates that the example provided is only one example of a possible scenario. Other scenarios may exist that could demonstrate the flow process for the web service.

Operations

cancel

This operation will cancel the input record based on the submission id.

Input Elements

- ❖ **SUBMISSION IDENTIFIER - Required.** This value will be the submission identifier that was previously passed back from the IDAssignmentV2 web service.
- ❖ **STATE STUDENT ID** - Should be left blank.
- ❖ **USER IDENTIFIER - Required.** This value will be a string that will identify the user ID that performed the action.
- ❖ **USER PASSWORD - Required.** This value will be a string that will identify the user password that performed the action.

Output Elements

Upon completion of the cancel request, the *NearMatchListResult element* is returned. The *NearMatchListResult element* will have the following properties:

- ❖ **Status** – this is a string containing the status of the function call. For a full list of status values refer to the NearMatchResolutionV2 Status Table below.

assignId

This operation will assign an already existing state student identifier to the input record.

Input Elements

- ❖ **SUBMISSION IDENTIFIER - Required.** This value will be the submission identifier that was previously passed back from the IDAssignmentV2 web service.
- ❖ **STATE STUDENT ID - Required.** The identifier selected to be assigned to the input record.
- ❖ **USER IDENTIFIER - Required.** This value will be a string that will identify the user ID that performed the action. Must be a Super User account.
- ❖ **USER PASSWORD - Required.** This value will be a string that will identify the user password that performed the action. Must be a Super User account password.

Output Elements

Upon completion of the assignment request, the *NearMatchListResult element* is returned. The *NearMatchListResult element* will have the following properties:

- ❖ **Status** – this is a string containing the status of the function call. For a full list of status values refer to the NearMatchResolutionV2 Status Table below.

createNewID

This operation will create a new state student identifier for the input record.

Input Elements

- ❖ **SUBMISSION IDENTIFIER - Required.** This value will be the submission identifier that was previously passed back from the IDAssignmentV2 web service.
- ❖ **STATE STUDENT ID** - Should be left blank.
- ❖ **USER IDENTIFIER - Required.** This value will be a string that will identify the user ID that performed the action. Must be a Super User account.
- ❖ **USER PASSWORD - Required.** This value will be a string that will identify the user password that performed the action. Must be a Super User account password.

Output Elements

Upon completion of the createNewId request, the *NearMatchListResult Object* is returned. The *NearMatchListResult element* will have the following properties:

- ❖ **Status** – this is a string containing the status of the function call. For a full list of status values refer to the NearMatchResolutionV2 Object Status Table below.

- ❖ *StateStudentId* – use this property to get the state id assigned to the student.
- ❖ *AliasID* – use this property to obtain the alias ID.

Reference WSDL

<https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2>

Shown in Appendix B.

Example: <https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2>

NearMatchResolutionV2 Object Status Table

Status Code	Description
ERROR	An exception occurred.
FAILED_USER_NOT_FOUND	The call failed because the user id was null.
FAILED_USER_NOT_FOUND_OR_INACTIVE	The call failed because the user id passed was not found in the system or the user id is not active.
FAILED_STAGE_RECORD_NOT_FOUND	The submission record cannot be found in the database.
FAILED_STAGE_RECORD_NOT_PENDING_NEAR_MATCH	The submission record is no longer pending the near match status.
FAILED_STATE_ID_NOT_FOUND	The state student identifier passed in was not found in the system.
FAILED_INVALID_INPUT_ID	The submission id is invalid.
SUCCESS	The call was successful.

StudentSearch – Searching For a Student / Staff

The Student Search Web Service will return a list of potential matches found by the TSDS Unique ID system for the input record submitted. This Web Service works similarly to the IDAssignmentV2 Web Services, but it does not assign identifiers in any case.

The Web Service will have one operation called searchStudent. Upon submission of a request, the Web Service will return a StudentSearchResult element. This element will have a status and an array of PotentialMatch elements. This array includes a match probability score that can be utilized to determine a match.

Note- records should not be considered matches if they are not higher than the Upper Near Match Threshold configured in the system. Records in a multiple match scenario should be reviewed before being accepted as a match. This Web Service is only functional with TSDS UID System 7 or above.

Operations

searchStudent

This operation will return potential match records that matched the submission record during the search request.

searchWithIDExactMatch

This operation will try to do an exact match with student demographic information when a valid STATE_STUDENT_ID is submitted in the STUDENTINPUT element. If an exact match is found, the result object containing single match record is returned. If exact match fails, result object will contain no matches. It behaves exactly like searchStudent operation when STATE_STUDENT_ID is empty or invalid. Unlike the regular Exact Match Rule, this does not return a list of potential matches. Refer to the STUDENTINPUT and output elements below.

Input Elements

- *STUDENTINPUT*-This element will represent the student to search. The student input elements are:

Field/Element	Description	Notes
alternateld	Optional Valid Values/Format: Length must be less than or equal to 50 characters.	
alternateSource	Optional Valid Values/Format: Length must be less than or equal to 60 characters.	
birthDate	Required. Valid Values/Format: yyyy-mm-dd; MM/DD/YYYY; Example: 2000-10-30	
cdfNumbersArray	Optional Valid Values/Format: Length must be less than or equal to 25 characters. Note-CDFs are not overwritten if subsequent submissions are blank. To overwrite CDFs NULL must be sent as the value.	

cdfValuesArray	Optional <i>Valid Values/Format:</i> Length must be less than or equal to 25 characters. To be used to identify the CDF number associated to the cdfValuesArray values passed. For example, if 3 CDF values are being passed, the cdfNumberArray should indicate which value should be assigned to the specific CDF numbers. CDFs range from 1 to 8.
districtCode	Required. <i>Valid Values/Format:</i> Must be valid based on user authentication and location.
gender	Required. <i>Valid Values/Format:</i> F or M

Field/Element	Description	Notes
gradeLevel	<p>Required. <i>Valid Values/Format:</i> EE (Early Education), PK (Pre-Kindergarten), KG (Kindergarten), 01 (Grade 1), 02 (Grade 2), 03 (Grade 3), 04 (Grade 4), 05 (Grade 5), 06 (Grade 6), 07 (Grade 7), 08 (Grade 8), 09 (Grade 9), 10 (Grade 10), 11 (Grade 11), 12 (Grade 12), and 00 (only valid for Staff)</p>	
localStudentId	<p>Required. <i>Valid Values/Format:</i> Length must be less than or equal to 20 characters. <i>Example:</i> 4857390022</p>	
previousLastName	<p>Optional. <i>Valid Values/Format:</i> Length must be less than or equal to 60 characters.</p>	
raceEthnicCode	<p>Required. <i>Valid Values/Format:</i> I A B P W</p>	
residentDistrictCode	<p>Optional. <i>Valid Values/Format:</i> Must be valid based on user authentication and location.</p>	
schoolCode	<p>Required. <i>Valid Values/Format:</i> Must be valid based on user authentication and district.</p>	
schoolYear	<p>Required. <i>Valid Values/Format:</i> Must be any 4 digit number, cannot start with 0. <i>Example:</i> 2008</p>	
sourceSISCode	<p>Required <i>Valid Values/Format:</i> Must validate to the available sources defined in the system.</p>	
SSN	<p>Required . <i>Valid Values/Format:</i> Must be 9 digits, can include “-” but must be formatted correctly with three, two and four digits. <i>Example:</i> 444335555 OR 444-33-5555 Or 8 digits with the character “S” in front. Example S44335555</p>	
stateStudentId	<p>Optional. <i>Valid Values/Format:</i> Must be numeric with a length of 10 or 12 characters.</p>	
studentFirstName	<p>Required. <i>Valid Values/Format:</i> Length must be less than or</p>	

Field/Element	Description	Notes
	equal to 60 characters.	
studentLastName	Required. <i>Valid Values/Format:</i> Length must be less than or equal to 60 characters.	
studentMiddleName	Optional. <i>Valid Values/Format:</i> Length must be less than or equal to 60 characters.	
studentSuffix	Optional. <i>Valid Values/Format:</i> 1 (Jr), 2 (Sr), 3 (II), 4 (III), 5 (IV), 6 (V), 7 (VI), 8 (VII), 9 (VIII), A (I), B (IX), C (X)	
ethnicityIndicator	Required <i>Valid Values/Format:</i> Length must be less than or equal to 4 characters.	
Race2Code	Optional <i>Valid Values/Format:</i> I A B P W	
Race3Code	Optional <i>Valid Values/Format:</i> I A B P W	
Race4Code	Optional <i>Valid Values/Format:</i> I A B P W	
Race5Code	Optional <i>Valid Values/Format:</i> I A B P W	

1. **USER ID - Required.** This value will be a string that will identify the user ID that performed the action. Must be a Super User account.
2. **USER PASSWORD - Required.** This value will be the user ID's password that performed the action. Must be a Super User account password.

Output Elements:

Upon completion of the StudentSearch request, the *StudentSearchResult* element is returned. The *StudentSearchResult* element will have the following properties:

1. *Status* – this is a string containing the status of the function call. For a full list of status values refer to the IDAssignmentV2 Status Table below.
2. *StateStudentId* – use this property to get the state id assigned to the student.
3. *SubmissionId* – use this property to get the submission id assigned to the transaction.
4. *BatchNumber* – use this property to get the batch number assigned to the transaction.
5. *cdfMap* – used internally for CDFs.
6. *aliasId* – this property will return the alias ID.
7. *PotentialNearMatches* – this property will return an array of PotentialMatch elements. The data that is returned in this array is for the near matching (master) records. Each near match element will have the following properties:

- a. STATE STUDENT ID (stateStudentID) – string with the state student Identifier of the matched record.
- b. LAST NAME (lastName) – the last name of the matched record.
- c. FIRST NAME (firstName) – the first name of the matched record.
- d. MIDDLE NAME (middleName) – the middle name of the matched record.
- e. SUFFIX (suffixName) – the suffix of the matched record.
- f. GENDER (gender) – the gender code of the matched record.
- g. DATE OF BIRTH (dob) – the date of birth of the matched record.
- h. ETHNIC CODE (ethnicCode) – the ethnic code of the matched record.
- i. SOCIAL SECURITY NUMBER (ssn) – the social security number of the matched record.
- j. MATCH PROBABILITY (matchProbability) – the match score ranges from Lower Near Match Threshold to Upper Near Match Threshold setting from System Properties.
- k. COMMENTS (comments) – the comments for the matched record.
- l. CREATION DATE (createDate) – the date the record was created.
- m. CLAIM REQUIRED (claimRequired) – indicates if a claim is required for the record. This can occur when Claiming is enabled.
- n. DISTRICT CODE (districtCode) – the District Code of the matched record.
- o. SCHOOL CODE (schoolCode) – the School Code of the matched record.
- p. SCHOOL YEAR (schoolYear) – the school year of the matched record.
- q. GRADE LEVEL (gradeLevel) – the grade level of the matched record.
- r. CDF NUMBERS ARRAY (cdfNumbersArray) – the array of CDF by number (V2 only)
- s. CDF VALUES ARRAY (cdfValuesArray) – the array of CDF values (V2 only)
- t. CDF REF NUMBER (cdfRefNumber) (V2 only) – the internal reference to the CDF.
- u. ALIAS ID (aliasID) (V2 only) – the alias ID of the matched record.
- v. ALTERNATE SOURCE (alternateSource) (V2 only) – the alternate source of the matched record.
- w. PREVIOUS LAST NAME (previousLastName) (V2 only) – the previous last name of the matched record.
- x. SOURCE SIS CODE (sourceSISCode) (V2 only) – the source system of the matched record.
- y. ETHNICITY INDICATOR (ethnicityIndicator) (V2 only) – the ethnic code for the matched record.
- z. RACE 2 CODE (Race2Code) (V2 only) – the race 2 code value for the matched record.
- aa. RACE 2 CODE Description (Race2CodeDesc) (V2 only) – the race 2 code value description for the matched record.
- bb. RACE 3 CODE (Race3Code) (V2 only) – the race 3 code value for the matched record.
- cc. RACE 3 CODE Description (Race3CodeDesc) (V2 only) – the race 3 code value description for the matched record
- dd. RACE 4 CODE (Race4Code) (V2 only) – the race 4 code value for the matched record.
- ee. RACE 4 CODE Description (Race4CodeDesc) (V2 only) – the race 4 code value description for the matched record.
- ff. RACE 5 CODE (Race5Code) (V2 only) – the race 5 code value for the matched record.

gg. RACE 5 CODE DESCRIPTION (Race2CodeDesc) (V2 only) – the race 5 code value description for the matched record

- *fieldsWithErrors* – returns an array of field errors. The possible messages are:
 - a. LAST_NAME – the submitted value was too long or null.
 - b. FIRST_NAME – the submitted value was too long or null.
 - c. MIDDLE_NAME – the submitted value was too long.
 - d. SUFFIX – the submitted value was too long.
 - e. GENDER – the submitted value was too long, invalid, or null.
 - f. DATE_OF_BIRTH – the submitted value was too long, invalid, or null.
 - g. ETHNICITY – the submitted value was too long or invalid.
 - h. GRADE_LEVEL - the submitted value was too long, invalid, or null.
 - i. LOCAL_STUDENT_ID – the submitted value was too long, invalid, or null.
 - j. SCHOOL_YEAR – the submitted value was too long, invalid, or null.
 - k. DISTRICT_CODE – the submitted value was too long, invalid, or null.
 - l. SCHOOL_CODE – the submitted value was too long, invalid, or null.
 - m. RES_DISTRICT_CODE – the submitted value was too long, invalid, or null.
 - n. SSN – the submitted value was invalid.
 - o. STATE_STUDENT_ID – the submitted value was too long or invalid.
 - p. SOURCE_SYSTEM – the submitted value was too long or invalid. (V2 only)
 - q. ALTERNATE_SOURCE – the submitted value was too long or invalid. (V2 only)
 - r. ALTERNATE_ID – the submitted value was invalid.
 - s. ALTERNATE_LAST_NAME - the submitted value was too long. (V2 only)
 - t. CDF_XXXX – the submitted value was too long (V2 only). XXXX stands for the name of the CDF with error.
 - u. AGENCY_NOT_FOUND – the submitted value was invalid.
 - v. AGENCY_NOT_AUTHORIZED – the user is not authorized to access this agency.

Reference WSDL

<https://tea4avdplbprod.tea.state.tx.us/uid/services/StudentSearch?W>

SDL Shown in Appendix B

Example: <https://tea4avdplbprod.tea.state.tx.us/uid/services/StudentSearch?WSDL>

Search Status Table

Status Code	Description
ERROR	An exception occurred.
FAILED_USER_NOT_FOUND _OR_INACTIVE	The call failed because the user id passed was not found in the system or the user id is not active.
SUCCESS	The call returned one or more potential match records.

FAILED_USER_NOT_FOUND	The call failed because the user id was null.
SUCCESS_MULTIPLE_MATCHING_RECORDS	The call was successful but multiple matches were found.
SUCCESS_SINGLE_MATCH_RECORD	The call was successful and a single match was found.
SUCCESS_NO_MATCHING_RECORDS	The call was successful and no match was found.
FAILED_VALIDATION	The submission failed data validation.
LOCKOUT	The submission failed data validation. When the system is locked out due to an Administrator lockout or forced lockout by a Matching Engine issue, the lockout status will be returned.

BatchInfoSearch – Searching For a Student / Staff

The Batch Info Search Web Service will return batch information for single record batches in TSDS Unique ID system for the input record submitted. TSDS Unique ID Version 7.0 or above is required for this Web Service.

The Web Service will have one operation called `getBatchInformation`. Upon submission of a request, the Web Service will return a `BatchInfoResult` element. This element will have a status and other batch information as described below.

Operations

`getBatchInformation`

This operation will return potential match records that matched the submission record during the search request.

Input Elements

1. *BATCHNUMBER*-This element will represent the batch number to search. It must be a valid number.
2. *USER ID - Required*. This value will be a string that will identify the user ID that performed the action. Must be a Super User account.
3. *USER PASSWORD - Required*. This value will be the user ID's password that performed the action. Must be a Super User account password.

Output Elements:

Upon completion of the `BatchInfo` request, the *BatchInfoResult element* is returned. The *BatchInfoResult* element will have the following properties:

1. *Status* – this is a string containing the status of the function call. For a full list of status values refer to the `IDAssignmentV2` Status Table below.
2. *StateStudentId* – use this property to get the state id assigned to the student.
3. *BatchNumber* – use this property to get the batch number assigned to the transaction.
4. *BatchStatus* – use this property to get the batch status for the requested batch. For batch status information, refer to the `Batch Status` table below.
5. *nearMatchCount* - use this property to get the total number of near match records in the batch.
6. *totalNumberRecords* – use this property to get the total number of records in the batch. This should always equal 1.
7. *transactionStatus* – use this property to get the transaction status of the record. Refer to the numeric values in the “`IDAssignmentV2 Transaction Status Table`” above.

Batch Statuses

Code	Status/Message
01	FILE UPLOAD COMPLETE
02	LOAD AND VALIDATION COMPLETE
03	ID CREATION STARTED
04	ID CREATION COMPLETE
05	ID CREATION ABORTED
06	ID EXTRACT COMPLETE
07	ARCHIVED TO HISTORY
08	CANCELED
09	INVALID

Code	Status/Message
10	LOAD AND VALIDATION STARTED
11	CLAIM STARTED (NOT USED BY TEA)
12	CLAIM COMPLETE (NOT USED BY TEA)
13	CLAIM CANCELED (NOT USED BY TEA)
14	SEARCH UPLOAD COMPLETE
15	SEARCH VALIDATION STARTED
16	SEARCH VALIDATION ABORTED
17	SEARCH VALIDATION COMPLETED
18	SEARCH STARTED
19	SEARCH ABORTED
20	SEARCH COMPLETED
21	VALIDATION INTERRUPTED
22	ID ASSIGNMENT INTERRUPTED
23	SEARCH VALIDATION INTERRUPTED
24	SEARCH INTERRUPTED
25	SEARCH DATA ERRORS PENDING
26	SEARCH canceled
27	FILE UPLOAD FAILED
28	DATA ERRORS PENDING
29	NEAR MATCHES PENDING

Note-The statuses show above include all possible Batch Statuses in the system. Not all of these will be encountered by Web Services.

Reference WSDL

<https://tea4avdplbprod.tea.state.tx.us/uid/services/BatchInfo?W>

SDL Shown in Appendix B.

Example: <https://tea4avdplbprod.tea.state.tx.us/uid/services/BatchInfo?WSDL>

BatchInfo Search Status Table

Status Code	Description
ERROR	An exception occurred.
FAILED_USER_NOT_FOUND_OR_INACTIVE	The call failed because the user id passed was not found in the system or the user id is not active.
SUCCESS	The call returned one or more potential match records.
FAILED_USER_NOT_FOUND	The call failed because the user id was null.
FAILED_BATCH_NOT_FOUND	The call failed because the batch submitted was not located in the system.
FAILED_BATCH_HAS_MULTIPLE_RECORDS	The call failed because there are multiple records in the batch submitted. The system only returns single record batches.
LOCKOUT	When the system is locked out due to an Administrator lockout or forced

Status Code	Description
FAILED_NO_TRANSACTION_ FOR_BATCH_NOT_FOUND	lockout by a Matching Engine issue, the lockout status will be returned. The call failed because there was no batch matching the submission batch.
FAILED_BATCH_HAS_NO_RECORD S	The call failed because the submitted batch did not have records.
FAILED_USER_NOT_AUTHORIZED	The call failed because the user was not authorized to submit the request.

Appendix A – Soap Message Examples

IDAssignmentV2 Request – New ID Example

The sample below demonstrates a SOAP request for a new ID assignment using the IDAssignmentV2 Web Service:

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:q0="http://com.escholar.ws"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns0:assignId xmlns:ns0="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <student xsi:type="q0:Student">
        <SSN xsi:type="xsd:string"/>
        <birthDate xsi:type="xsd:string">05/01/1988</birthDate>
        <districtCode xsi:type="xsd:string">123456</districtCode>
        <gender xsi:type="xsd:string">M</gender>
        <gradeLevel xsi:type="xsd:string">10</gradeLevel>
        <localStudentId xsi:type="xsd:string">1113322</localStudentId>
        <raceEthnicCode xsi:type="xsd:string">W</raceEthnicCode>
        <residentDistrictCode xsi:type="xsd:string">0009</residentDistrictCode>
        <schoolCode xsi:type="xsd:string">123456789</schoolCode>
        <schoolYear xsi:type="xsd:string">2007</schoolYear>
        <stateStudentId xsi:type="xsd:string"/>
        <studentFirstName xsi:type="xsd:string">Mira</studentFirstName>
        <studentLastName xsi:type="xsd:string">Rauen</studentLastName>
        <studentMiddleName xsi:type="xsd:string"/>
        <studentSuffix xsi:type="xsd:string"/>
      </student>
      <userId xsi:type="xsd:string">*****</userId>
      <password xsi:type="xsd:string">*****</password>
    </ns0:assignId>
  </soapenv:Body>
</soapenv:Envelope>
```

IDAssignmentV2 Response – New ID Example

The sample below demonstrates the SOAP response to the above IDAssignmentV2 submission. The request did not include any data issues and there were no matches, so the response provided the new ID for assignment.

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:assignIdResponse xmlns:ns1="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">
      <assignIdReturn href="#id0"/>
    </ns1:assignIdResponse>
    <multiRef xmlns:ns2="http://com.escholar.ws"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:IDAssignmentResultV2">
      <batchNumber xsi:type="xsd:string">266</batchNumber>
      <fieldsWithErrors xsi:nil="true" xsi:type="soapenc:Array"/>
      <potentialNearMatches xmlns:ns3="http://www.escholar.com"
        xsi:nil="true" xsi:type="ns3:NearMatchArray"/>
      <stateStudentId xsi:type="xsd:string">5202763107</stateStudentId>
      <status xsi:type="xsd:string">05</status>
      <submissionId xsi:type="xsd:string">1164</submissionId>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

IDAssignmentV2 Request – Match List Found Example

The sample below demonstrates a SOAP request for a new ID assignment using the IDAssignmentV2 Web Service where matching records **were** found:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:q0="http://com.escholar.ws" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns0:assignId xmlns:ns0="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">
      <student xsi:type="q0:Student">
        <SSN xsi:type="xsd:string"/>
        <birthDate xsi:type="xsd:string">05/01/1988</birthDate>
        <districtCode xsi:type="xsd:string">123456</districtCode>
        <gender xsi:type="xsd:string">M</gender>
        <gradeLevel xsi:type="xsd:string">10</gradeLevel>
        <localStudentId xsi:type="xsd:string">3433322</localStudentId>
        <raceEthnicCode xsi:type="xsd:string">W</raceEthnicCode>
        <residentDistrictCode xsi:type="xsd:string">0009</residentDistrictCode>
        <schoolCode xsi:type="xsd:string">123456789</schoolCode>
        <schoolYear xsi:type="xsd:string">2007</schoolYear>
        <stateStudentId xsi:type="xsd:string"/>
        <studentFirstName xsi:type="xsd:string">Michael</studentFirstName>
        <studentLastName xsi:type="xsd:string">Jordin</studentLastName>
        <studentMiddleName xsi:type="xsd:string"/>
        <studentSuffix xsi:type="xsd:string"/>
      </student>
      <userId xsi:type="xsd:string">****</userId>
      <password xsi:type="xsd:string">****</password>
    </ns0:assignId>
  </soapenv:Body>
</soapenv:Envelope>
```

IDAssignmentV2 Response – Match List Found Example

The sample below demonstrates the SOAP response to the above IDAssignmentV2 Match List Found sample. The request did not include any data issues and but did include matches. The near match list was returned, including a submission ID:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:assignIdResponse xmlns:ns1="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <multiRef xmlns:ns2="http://com.escholar.ws"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0"
        soapenc:root="0"
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xsi:type="ns2:IDAssignmentResultV2">
        <batchNumber xsi:type="xsd:string">265</batchNumber>
        <fieldsWithErrors xsi:nil="true" xsi:type="soapenc:Array"/>
        <potentialNearMatches soapenc:arrayType="ns2:PotentialMatch[2]"
          xsi:type="soapenc:Array">
          <potentialNearMatches href="#id1"/>
          <potentialNearMatches href="#id2"/>
        </potentialNearMatches>
        <stateStudentId xsi:nil="true" xsi:type="xsd:string"/>
        <status xsi:type="xsd:string">06</status>
        <submissionId xsi:type="xsd:string">1163</submissionId>
      </multiRef>
      <multiRef xmlns:ns3="http://com.escholar.ws"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id2"
        soapenc:root="0"
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xsi:type="ns3:PotentialMatch">
        <claimRequired xsi:type="xsd:boolean">>false</claimRequired>
        <comments xsi:type="xsd:string">null</comments>
        <creationDate xsi:type="xsd:dateTime">2007-11-
          15T13:21:19.000Z</creationDate>
        <districtCode xsi:type="xsd:string">123456</districtCode>
        <dob xsi:type="xsd:dateTime">1984-05-01T04:00:00.000Z</dob>
        <ethnicCode xsi:type="xsd:string">A</ethnicCode>
        <firstName xsi:type="xsd:string">MICHAEL</firstName>
        <gender xsi:type="xsd:string">M</gender>
        <gradeLevel xsi:type="xsd:string">03</gradeLevel>
        <lastName xsi:type="xsd:string">JORDIN</lastName>
        <matchProbability xsi:type="xsd:double">89.0</matchProbability>
        <middleName xsi:type="xsd:string">R</middleName>
        <relationType xsi:type="xsd:string">0</relationType>
        <schoolCode xsi:type="xsd:string">123456789</schoolCode>
        <schoolYear xsi:type="xsd:string">2006</schoolYear>
        <ssn xsi:type="xsd:string">863261269</ssn>
        <stateStudentID xsi:type="xsd:string">6039801123</stateStudentID>
        <suffixName xsi:type="xsd:string">JR</suffixName>
      </multiRef>
      <multiRef xmlns:ns4="http://com.escholar.ws"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id1"
        soapenc:root="0"
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xsi:type="ns4:PotentialMatch">
```



```

    <claimRequired xsi:type="xsd:boolean">false</claimRequired>
    <comments xsi:type="xsd:string">null</comments>
    <creationDate xsi:type="xsd:dateTime">2007-11-
      15T13:21:19.000Z</creationDate>
    <districtCode xsi:type="xsd:string">123456</districtCode>
    <dob xsi:type="xsd:dateTime">1999-05-01T04:00:00.000Z</dob>
    <ethnicCode xsi:type="xsd:string">A</ethnicCode>
    <firstName xsi:type="xsd:string">MICHAEL</firstName>
    <gender xsi:type="xsd:string">M</gender>
    <gradeLevel xsi:type="xsd:string">04</gradeLevel>
    <lastName xsi:type="xsd:string">JORDIN</lastName>
    <matchProbability xsi:type="xsd:double">89.0</matchProbability>
    <middleName xsi:type="xsd:string">R</middleName>
    <relationType xsi:type="xsd:string">0</relationType>
    <schoolCode xsi:type="xsd:string">123456789</schoolCode>
    <schoolYear xsi:type="xsd:string">2007</schoolYear>
    <ssn xsi:type="xsd:string">863261269</ssn>
    <stateStudentID xsi:type="xsd:string">2436954461</stateStudentID>
    <suffixName xsi:type="xsd:string">null</suffixName>
  </multiRef>
</ns1:assignIdResponse>
</soapenv:Body>
</soapenv:Envelope>

```

NearMatchListV2 Request - Existing Near Match List

The sample below demonstrates a SOAP request for a Near Match List using the NearMatchListV2 Web Service. This example uses the submission information listed the “IDAssignmentV2 Response – Match List Found Example” above.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns0:getNearMatches xmlns:ns0="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <submissionId xsi:type="xsd:string">1163</submissionId>
      <userId xsi:type="xsd:string">****</userId>
      <password xsi:type="xsd:string">****</password>
    </ns0:getNearMatches>
  </soapenv:Body>
</soapenv:Envelope>

```

NearMatchListV2 Response – Existing Near Match List

This sample demonstrates the SOAP response for the above NearMatchListV2 request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getNearMatchesResponse xmlns:ns1="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <getNearMatchesReturn href="#id0"/>
    </ns1:getNearMatchesResponse>
    <multiRef xmlns:ns2="http://www.escholar.com"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0" soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:NearMatchListResult">
      <potentialNearMatches soapenc:arrayType="ns2:PotentialMatch[2]"
        xsi:type="soapenc:Array">
        <potentialNearMatches href="#id1"/>
        <potentialNearMatches href="#id2"/>
      </potentialNearMatches>
      <status xsi:type="xsd:string">SUCCESS</status>
    </multiRef>
    <multiRef xmlns:ns3="http://www.escholar.com"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id1" soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns3:PotentialMatch">
      <claimRequired xsi:type="xsd:boolean">>false</claimRequired>
      <comments xsi:type="xsd:string">null</comments>
      <creationDate xsi:type="xsd:dateTime">2007-11-15T13:21:19.000Z</creationDate>
      <districtCode xsi:type="xsd:string">123456</districtCode>
      <dob xsi:type="xsd:dateTime">1999-05-01T04:00:00.000Z</dob>
      <ethnicCode xsi:type="xsd:string">A</ethnicCode>
      <firstName xsi:type="xsd:string">MICHAEL</firstName>
      <gender xsi:type="xsd:string">M</gender>
      <gradeLevel xsi:type="xsd:string">04</gradeLevel>
      <lastName xsi:type="xsd:string">JORDIN</lastName>
      <matchProbability xsi:type="xsd:double">89.0</matchProbability>
      <middleName xsi:type="xsd:string">R</middleName>
      <relationType xsi:type="xsd:string">0</relationType>
      <schoolCode xsi:type="xsd:string">123456789</schoolCode>
      <schoolYear xsi:type="xsd:string">2007</schoolYear>
      <ssn xsi:type="xsd:string">863261269</ssn>
      <stateStudentID xsi:type="xsd:string">2436954461</stateStudentID>
      <suffixName xsi:type="xsd:string">null</suffixName>
    </multiRef>
    <multiRef xmlns:ns4="http://www.escholar.com"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id2" soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns4:PotentialMatch">
      <claimRequired xsi:type="xsd:boolean">>false</claimRequired>
      <comments xsi:type="xsd:string">null</comments>
      <creationDate xsi:type="xsd:dateTime">2007-11-
        15T13:21:19.000Z</creationDate>
      <districtCode xsi:type="xsd:string">123456</districtCode>
      <dob xsi:type="xsd:dateTime">1984-05-01T04:00:00.000Z</dob>
      <ethnicCode xsi:type="xsd:string">A</ethnicCode>
      <firstName xsi:type="xsd:string">MICHAEL</firstName>
```

```
<gender xsi:type="xsd:string">M</gender>
<gradeLevel xsi:type="xsd:string">03</gradeLevel>
<lastName xsi:type="xsd:string">JORDIN</lastName>
<matchProbability xsi:type="xsd:double">89.0</matchProbability>
<middleName xsi:type="xsd:string">R</middleName>
<relationType xsi:type="xsd:string">0</relationType>
<schoolCode xsi:type="xsd:string">123456789</schoolCode>
<schoolYear xsi:type="xsd:string">2006</schoolYear>
<ssn xsi:type="xsd:string">863261269</ssn>
<stateStudentID xsi:type="xsd:string">6039801123</stateStudentID>
<suffixName xsi:type="xsd:string">JR</suffixName>
</multiRef>
</soapenv:Body>
</soapenv:Envelope>
```

NearMatchResolutionV2 Request – Create a New ID

This sample demonstrates a SOAP request to create a new ID from a near match list.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns0:assignId xmlns:ns0="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <tranSerialNumber xsi:type="xsd:string">1184</tranSerialNumber>
      <stateStudentId xsi:type="xsd:string">6241478796</stateStudentId>
      <userId xsi:type="xsd:string">****</userId>
      <password xsi:type="xsd:string">****</password>
    </ns0:assignId>
  </soapenv:Body>
</soapenv:Envelope>
```

NearMatchResolutionV2 Response – Create a New ID

This sample demonstrates a SOAP response to create the new ID request above.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns1:assignIdResponse xmlns:ns1="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <assignIdReturn href="#id0"/>
    </ns1:assignIdResponse>
    <multiRef xmlns:ns2="http://www.escholar.com"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0" soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:NearMatchResolutionResult">
      <stateStudentId xsi:type="xsd:string">6241478796</stateStudentId>
      <status xsi:type="xsd:string">SUCCESS</status>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

NearMatchResolutionV2 Request – Assign ID

This sample demonstrates the SOAP request for the NearMatchResolutionV2 service for a submission ID that does not have pending near matches:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns0:assignId xmlns:ns0="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <tranSerialNumber xsi:type="xsd:string">1163</tranSerialNumber>
      <stateStudentId xsi:type="xsd:string"/>
      <userId xsi:type="xsd:string">****</userId>
      <password xsi:type="xsd:string">****</password>
    </ns0:assignId>
  </soapenv:Body>
</soapenv:Envelope>
```

NearMatchResolutionV2 Response – Assign ID with Error

This sample demonstrates the SOAP response for the above no pending matches request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns1:assignIdResponse xmlns:ns1="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <assignIdReturn href="#id0"/>
    </ns1:assignIdResponse>
    <multiRef xmlns:ns2="http://www.escholar.com"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0" soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:NearMatchResolutionResult">
      <stateStudentId xsi:nil="true" xsi:type="xsd:string"/>
      <status xsi:type="xsd:string">FAILED_STAGE_RECORD_NOT_PENDING_NEAR_MAT
      CH</status>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

NearMatchResolutionV2 Request – Cancel

This sample demonstrates a SOAP request to cancel a near match using the submission ID.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns0:cancel xmlns:ns0="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <tranSerialNumber xsi:type="xsd:string">1163</tranSerialNumber>
      <userId xsi:type="xsd:string">****</userId>
      <password xsi:type="xsd:string">****</password>
    </ns0:cancel>
  </soapenv:Body>
</soapenv:Envelope>
```

NearMatchResolutionV2 Response – Cancel

This sample demonstrates the SOAP response to a NearMatchResolutionV2 cancel request using the information from the “NearMatchResolutionV2 Request – Cancel” from above.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns1:cancelResponse xmlns:ns1="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <cancelReturn href="#id0"/>
    </ns1:cancelResponse>
    <multiRef xmlns:ns2="http://www.escholar.com"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0" soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:NearMatchResolutionResult">
      <stateStudentId xsi:nil="true" xsi:type="xsd:string"/>
      <status xsi:type="xsd:string">SUCCESS</status>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

Error Response Samples – Invalid Username/Password

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <soapenv:Body>
    <ns1:assignIdResponse xmlns:ns1="http://ws.escholar.com"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <assignIdReturn href="#id0"/>
    </ns1:assignIdResponse>
    <multiRef xmlns:ns2="http://com.escholar.ws"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0" soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:IDAssignmentResultV2">
      <batchNumber xsi:nil="true" xsi:type="xsd:string"/>
      <fieldsWithErrors xsi:nil="true" xsi:type="soapenc:Array"/>
      <potentialNearMatches xmlns:ns3="http://www.escholar.com" xsi:nil="true"
        xsi:type="ns3:NearMatchArray"/>
      <stateStudentId xsi:nil="true" xsi:type="xsd:string"/>
      <status xsi:type="xsd:string">FAILED_USER_NOT_FOUND_OR_INACTIVE</status>
      <submissionId xsi:nil="true" xsi:type="xsd:string"/>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

******NOTE: Username and password for above request statements not included for security purposes.**

Appendix B – WSDL for Web Services

BatchInfo WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/BatchInfo"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="https://tea4avdplbprod.tea.state.tx.us/uid/services/BatchInfo"
xmlns:intf="https://tea4avdplbprod.tea.state.tx.us/uid/services/BatchInfo"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="http://www.escholar.com"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--
    WSDL created by Apache Axis version: 1.4
    Built on Apr 22, 2006 (06:55:48 PDT)
  -->
  <wsdl:types>
    <schema targetNamespace="http://www.escholar.com"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="BatchInfoResult">
        <sequence>
          <element name="batchNumber" type="xsd:long"/>
          <element name="batchStatus" nillable="true" type="xsd:string"/>
          <element name="nearMatchCount" type="xsd:long"/>
          <element name="stateStudentId" nillable="true" type="xsd:string"/>
          <element name="status" nillable="true" type="xsd:string"/>
          <element name="totalNumberRecords" type="xsd:long"/>
          <element name="transactionStatus" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>
  <wsdl:message name="getBatchInformationResponse">
    <wsdl:part name="getBatchInformationReturn" type="tns1:BatchInfoResult"/>
  </wsdl:message>
  <wsdl:message name="getBatchInformationRequest">
    <wsdl:part name="batchNumber" type="xsd:long"/>
    <wsdl:part name="userId" type="xsd:string"/>
    <wsdl:part name="password" type="xsd:string"/>
  </wsdl:message>
  <wsdl:portType name="BatchInfo">
    <wsdl:operation name="getBatchInformation" parameterOrder="batchNumber userId password">
      <wsdl:input message="impl:getBatchInformationRequest"
name="getBatchInformationRequest"/>
      <wsdl:output message="impl:getBatchInformationResponse"
name="getBatchInformationResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="BatchInfoSoapBinding" type="impl:BatchInfo">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getBatchInformation">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getBatchInformationRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://ws.escholar.com" use="encoded"/>
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

```
        <wsdl:output name="getBatchInformationResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/BatchInfo"
                use="encoded"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="BatchInfoService">
    <wsdl:port binding="impl:BatchInfoSoapBinding" name="BatchInfo">
        <wsdlsoap:address
location="https://tea4avdplbprod.tea.state.tx.us/uid/services/BatchInfo"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```


IDAssignmentV2 WSDL, operation assigned

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2"
xmlns:apacheSOAP="http://xml.apache.org/xml-soap"
xmlns:impl="https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2"
xmlns:intf="https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="http://www.escholar.com"
xmlns:tns2="http://util.java" xmlns:tns3="http://sql.java" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--
    WSDL created by Apache Axis version: 1.4
    Built on Apr 22, 2006 (06:55:48 PDT)
  -->
  <wsdl:types>
    <schema
targetNamespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://sql.java"/>
      <import namespace="http://util.java"/>
      <import namespace="http://www.escholar.com"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="ArrayOf_xsd_string">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
wsdl:arrayType="xsd:string[]" />
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://www.escholar.com"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://sql.java"/>
      <import namespace="http://util.java"/>
      <import
namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="StudentV2">
        <sequence>
          <element name="SSN" nillable="true" type="xsd:string"/>
          <element name="alternateId" nillable="true" type="xsd:string"/>
          <element name="alternateSource" nillable="true" type="xsd:string"/>
          <element name="birthDate" nillable="true" type="xsd:string"/>
          <element name="cdfNumbersArray" nillable="true"
type="impl:ArrayOf_xsd_string"/>
          <element name="cdfValuesArray" nillable="true"
type="impl:ArrayOf_xsd_string"/>
          <element name="districtCode" nillable="true" type="xsd:string"/>
          <element name="ethnicityIndicator" nillable="true" type="xsd:string"/>
          <element name="gender" nillable="true" type="xsd:string"/>
          <element name="gradeLevel" nillable="true" type="xsd:string"/>
          <element name="localStudentId" nillable="true" type="xsd:string"/>
          <element name="multiracialBinarySum" nillable="true"
type="xsd:double"/>
          <element name="previousLastName" nillable="true" type="xsd:string"/>
          <element name="race2Code" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>

```

```

<element name="race3Code" nillable="true" type="xsd:string"/>
<element name="race4Code" nillable="true" type="xsd:string"/>
<element name="race5Code" nillable="true" type="xsd:string"/>
<element name="raceEthnicCode" nillable="true" type="xsd:string"/>
<element name="recordUpdateDate" nillable="true" type="xsd:string"/>
<element name="recordUpdateDt" nillable="true" type="xsd:dateTime"/>
<element name="residentDistrictCode" nillable="true" type="xsd:string"/>
<element name="schoolCode" nillable="true" type="xsd:string"/>
<element name="schoolYear" nillable="true" type="xsd:string"/>
<element name="sourceSISCode" nillable="true" type="xsd:string"/>
<element name="stateStudentId" nillable="true" type="xsd:string"/>
<element name="studentFirstName" nillable="true" type="xsd:string"/>
<element name="studentLastName" nillable="true" type="xsd:string"/>
<element name="studentMiddleName" nillable="true" type="xsd:string"/>
<element name="studentSuffix" nillable="true" type="xsd:string"/>
</sequence>
</complexType>
<complexType name="PotentialMatchV2">
<sequence>
<element name="aliasId" nillable="true" type="xsd:string"/>
<element name="alternateSource" nillable="true" type="xsd:string"/>
<element name="cdfNumbersArray" nillable="true"
type="impl:ArrayOf_xsd_string"/>
<element name="cdfRefNumber" nillable="true" type="xsd:decimal"/>
<element name="cdfValuesArray" nillable="true"
type="impl:ArrayOf_xsd_string"/>
<element name="claimRequired" type="xsd:boolean"/>
<element name="comments" nillable="true" type="xsd:string"/>
<element name="creationDate" nillable="true" type="xsd:anyType"/>
<element name="districtCode" nillable="true" type="xsd:string"/>
<element name="dob" nillable="true" type="xsd:dateTime"/>
<element name="ethnicCode" nillable="true" type="xsd:string"/>
<element name="ethnicCodeDesc" nillable="true" type="xsd:string"/>
<element name="ethnicityIndicator" nillable="true" type="xsd:string"/>
<element name="ethnicityIndicatorDesc" nillable="true"
type="xsd:string"/>
<element name="firstName" nillable="true" type="xsd:string"/>
<element name="gender" nillable="true" type="xsd:string"/>
<element name="gradeLevel" nillable="true" type="xsd:string"/>
<element name="lastName" nillable="true" type="xsd:string"/>
<element name="matchProbability" type="xsd:double"/>
<element name="middleName" nillable="true" type="xsd:string"/>
<element name="multiracialBinarySum" nillable="true"
type="xsd:double"/>
<element name="previousLastName" nillable="true" type="xsd:string"/>
<element name="race2Code" nillable="true" type="xsd:string"/>
<element name="race2CodeDesc" nillable="true" type="xsd:string"/>
<element name="race3Code" nillable="true" type="xsd:string"/>
<element name="race3CodeDesc" nillable="true" type="xsd:string"/>
<element name="race4Code" nillable="true" type="xsd:string"/>
<element name="race4CodeDesc" nillable="true" type="xsd:string"/>
<element name="race5Code" nillable="true" type="xsd:string"/>
<element name="race5CodeDesc" nillable="true" type="xsd:string"/>
<element name="relationType" nillable="true" type="xsd:string"/>
<element name="schoolCode" nillable="true" type="xsd:string"/>
<element name="schoolYear" nillable="true" type="xsd:string"/>
<element name="sourceSISCode" nillable="true" type="xsd:string"/>

```

```

    <element name="ssn" nillable="true" type="xsd:string"/>
    <element name="stateStudentID" nillable="true" type="xsd:string"/>
    <element name="suffixName" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
<complexType name="NearMatchArray">
  <complexContent>
    <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType"
wsdl:arrayType="tns1:PotentialMatchV2[]" />
    </restriction>
  </complexContent>
</complexType>
<complexType name="IDAssignmentResultV2">
  <sequence>
    <element name="aliasId" nillable="true" type="xsd:string"/>
    <element name="batchNumber" nillable="true" type="xsd:string"/>
    <element name="cdfMap" nillable="true" type="xsd:anyType"/>
    <element name="comments" nillable="true" type="xsd:string"/>
    <element name="fieldsWithErrors" nillable="true"
type="impl:ArrayOf_xsd_string"/>
    <element name="potentialNearMatches" nillable="true"
type="tns1:NearMatchArray"/>
    <element name="stateStudentId" nillable="true" type="xsd:string"/>
    <element name="status" nillable="true" type="xsd:string"/>
    <element name="submissionId" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
</schema>
</wsdl:types>
<wsdl:message name="assignIdRequest">
  <wsdl:part name="student" type="tns1:StudentV2"/>
  <wsdl:part name="userId" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="assignIdResponse">
  <wsdl:part name="assignIdReturn" type="tns1:IDAssignmentResultV2"/>
</wsdl:message>
<wsdl:portType name="IDAssignmentV2">
  <wsdl:operation name="assignId" parameterOrder="student userId password">
    <wsdl:input message="impl:assignIdRequest" name="assignIdRequest"/>
    <wsdl:output message="impl:assignIdResponse" name="assignIdResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="IDAssignmentV2SoapBinding" type="impl:IDAssignmentV2">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="assignId">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="assignIdRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://wsv2.escholar.com" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="assignIdResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="https://tea4avdp1bprod.tea.state.tx.us/uid/services/IDAssignmentV2" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>

```

```
</wsdl:binding>  
<wsdl:service name="IDAssignmentV2Service">  
  <wsdl:port binding="impl:IDAssignmentV2SoapBinding" name="IDAssignmentV2">  
    <wsdlsoap:address  
location="https://tea4avdplbprod.tea.state.tx.us/uid/services/IDAssignmentV2"/>  
  </wsdl:port>  
</wsdl:service>  
</wsdl:definitions>
```

NearMatchListV2 WSDL, operation getNearMatches

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2"
xmlns:intf="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="http://www.escholar.com"
xmlns:tns2="http://sql.java" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--
    WSDL created by Apache Axis version: 1.4
    Built on Apr 22, 2006 (06:55:48 PDT)
  -->
  <wsdl:types>
    <schema
targetNamespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://sql.java"/>
      <import namespace="http://www.escholar.com"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="ArrayOf_xsd_string">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
wsdl:arrayType="xsd:string[]" />
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://www.escholar.com"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://sql.java"/>
      <import
namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="PotentialMatchV2">
        <sequence>
          <element name="aliasId" nillable="true" type="xsd:string"/>
          <element name="alternateSource" nillable="true" type="xsd:string"/>
          <element name="cdfNumbersArray" nillable="true"
type="impl:ArrayOf_xsd_string"/>
          <element name="cdfRefNumber" nillable="true" type="xsd:decimal"/>
          <element name="cdfValuesArray" nillable="true"
type="impl:ArrayOf_xsd_string"/>
          <element name="claimRequired" type="xsd:boolean"/>
          <element name="comments" nillable="true" type="xsd:string"/>
          <element name="creationDate" nillable="true" type="xsd:anyType"/>
          <element name="districtCode" nillable="true" type="xsd:string"/>
          <element name="dob" nillable="true" type="xsd:dateTime"/>
          <element name="ethnicCode" nillable="true" type="xsd:string"/>
          <element name="ethnicCodeDesc" nillable="true" type="xsd:string"/>
          <element name="ethnicityIndicator" nillable="true" type="xsd:string"/>
          <element name="ethnicityIndicatorDesc" nillable="true"
type="xsd:string"/>
          <element name="firstName" nillable="true" type="xsd:string"/>
          <element name="gender" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>

```

```

<element name="gradeLevel" nillable="true" type="xsd:string"/>
<element name="lastName" nillable="true" type="xsd:string"/>
<element name="matchProbability" type="xsd:double"/>
<element name="middleName" nillable="true" type="xsd:string"/>
<element name="multiracialBinarySum" nillable="true"
type="xsd:double"/>
<element name="previousLastName" nillable="true" type="xsd:string"/>
<element name="race2Code" nillable="true" type="xsd:string"/>
<element name="race2CodeDesc" nillable="true" type="xsd:string"/>
<element name="race3Code" nillable="true" type="xsd:string"/>
<element name="race3CodeDesc" nillable="true" type="xsd:string"/>
<element name="race4Code" nillable="true" type="xsd:string"/>
<element name="race4CodeDesc" nillable="true" type="xsd:string"/>
<element name="race5Code" nillable="true" type="xsd:string"/>
<element name="race5CodeDesc" nillable="true" type="xsd:string"/>
<element name="relationType" nillable="true" type="xsd:string"/>
<element name="schoolCode" nillable="true" type="xsd:string"/>
<element name="schoolYear" nillable="true" type="xsd:string"/>
<element name="sourceSISCode" nillable="true" type="xsd:string"/>
<element name="ssn" nillable="true" type="xsd:string"/>
<element name="stateStudentID" nillable="true" type="xsd:string"/>
<element name="suffixName" nillable="true" type="xsd:string"/>
</sequence>
</complexType>
<complexType name="NearMatchArrayV2">
<complexContent>
<restriction base="soapenc:Array">
<attribute ref="soapenc:arrayType"
wsdl:arrayType="tns1:PotentialMatchV2[]"/>
</restriction>
</complexContent>
</complexType>
<complexType name="NearMatchListResultV2">
<sequence>
<element name="potentialNearMatches" nillable="true"
type="tns1:NearMatchArrayV2"/>
<element name="status" nillable="true" type="xsd:string"/>
</sequence>
</complexType>
</schema>
</wsdl:types>
<wsdl:message name="getNearMatchesRequest">
<wsdl:part name="submissionId" type="xsd:string"/>
<wsdl:part name="userId" type="xsd:string"/>
<wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getNearMatchesResponse">
<wsdl:part name="getNearMatchesReturn" type="tns1:NearMatchListResultV2"/>
</wsdl:message>
<wsdl:portType name="NearMatchListV2">
<wsdl:operation name="getNearMatches" parameterOrder="submissionId userId password">
<wsdl:input message="impl:getNearMatchesRequest"
name="getNearMatchesRequest"/>
<wsdl:output message="impl:getNearMatchesResponse"
name="getNearMatchesResponse"/>
</wsdl:operation>
</wsdl:portType>

```

```
<wsdl:binding name="NearMatchListV2SoapBinding" type="impl:NearMatchListV2">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getNearMatches">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getNearMatchesRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://wsv2.escholar.com" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="getNearMatchesResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="NearMatchListV2Service">
  <wsdl:port binding="impl:NearMatchListV2SoapBinding" name="NearMatchListV2">
    <wsdlsoap:address
location="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchListV2"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

NearMatchResolutionV2 WSDL, operations: cancel, assignId, createNewId

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2"
xmlns:intf="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="http://www.escholar.com"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--
    WSDL created by Apache Axis version: 1.4
    Built on Apr 22, 2006 (06:55:48 PDT)
  -->
  <wsdl:types>
    <schema targetNamespace="http://www.escholar.com"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://xml.apache.org/xml-soap"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="NearMatchResolutionResultV2">
        <sequence>
          <element name="aliasId" nillable="true" type="xsd:string"/>
          <element name="stateStudentId" nillable="true" type="xsd:string"/>
          <element name="status" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
    </schema>
    <schema targetNamespace="http://xml.apache.org/xml-soap"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://www.escholar.com"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="mapItem">
        <sequence>
          <element name="key" nillable="true" type="xsd:anyType"/>
          <element name="value" nillable="true" type="xsd:anyType"/>
        </sequence>
      </complexType>
      <complexType name="Map">
        <sequence>
          <element maxOccurs="unbounded" minOccurs="0" name="item"
type="apachesoap:mapItem"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>
  <wsdl:message name="assignIdResponse">
    <wsdl:part name="assignIdReturn" type="tns1:NearMatchResolutionResultV2"/>
  </wsdl:message>
  <wsdl:message name="cancelRequest">
    <wsdl:part name="tranSerialNumber" type="xsd:string"/>
    <wsdl:part name="stateStudentId" type="xsd:string"/>
    <wsdl:part name="userId" type="xsd:string"/>
    <wsdl:part name="password" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="assignIdRequest">
    <wsdl:part name="tranSerialNumber" type="xsd:string"/>
    <wsdl:part name="stateStudentId" type="xsd:string"/>
  </wsdl:message>
</wsdl:definitions>
```



```

        <wsdl:part name="userId" type="xsd:string"/>
        <wsdl:part name="password" type="xsd:string"/>
    </wsdl:message>
    <wsdl:message name="auditResolveNearMatchEntryRequest">
        <wsdl:part name="msg" type="xsd:string"/>
        <wsdl:part name="loginId" type="xsd:string"/>
        <wsdl:part name="p" type="apachesoap:Map"/>
    </wsdl:message>
    <wsdl:message name="cancelResponse">
        <wsdl:part name="cancelReturn" type="tns1:NearMatchResolutionResultV2"/>
    </wsdl:message>
    <wsdl:message name="createNewIdResponse">
        <wsdl:part name="createNewIdReturn" type="tns1:NearMatchResolutionResultV2"/>
    </wsdl:message>
    <wsdl:message name="createNewIdRequest">
        <wsdl:part name="tranSerialNumber" type="xsd:string"/>
        <wsdl:part name="stateStudentId" type="xsd:string"/>
        <wsdl:part name="userId" type="xsd:string"/>
        <wsdl:part name="password" type="xsd:string"/>
    </wsdl:message>
    <wsdl:message name="auditResolveNearMatchEntryResponse"></wsdl:message>
    <wsdl:portType name="NearMatchResolutionV2">
        <wsdl:operation name="cancel" parameterOrder="tranSerialNumber stateStudentId userId
password">
            <wsdl:input message="impl:cancelRequest" name="cancelRequest"/>
            <wsdl:output message="impl:cancelResponse" name="cancelResponse"/>
        </wsdl:operation>
        <wsdl:operation name="assignId" parameterOrder="tranSerialNumber stateStudentId userId
password">
            <wsdl:input message="impl:assignIdRequest" name="assignIdRequest"/>
            <wsdl:output message="impl:assignIdResponse" name="assignIdResponse"/>
        </wsdl:operation>
        <wsdl:operation name="createNewId" parameterOrder="tranSerialNumber stateStudentId userId
password">
            <wsdl:input message="impl:createNewIdRequest" name="createNewIdRequest"/>
            <wsdl:output message="impl:createNewIdResponse" name="createNewIdResponse"/>
        </wsdl:operation>
        <wsdl:operation name="auditResolveNearMatchEntry" parameterOrder="msg loginId p">
            <wsdl:input message="impl:auditResolveNearMatchEntryRequest"
name="auditResolveNearMatchEntryRequest"/>
            <wsdl:output message="impl:auditResolveNearMatchEntryResponse"
name="auditResolveNearMatchEntryResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="NearMatchResolutionV2SoapBinding" type="impl:NearMatchResolutionV2">
        <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="cancel">
            <wsdlsoap:operation soapAction=""/>
            <wsdl:input name="cancelRequest">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://wsv2.escholar.com" use="encoded"/>
            </wsdl:input>
            <wsdl:output name="cancelResponse">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2" use="encoded"/>
            </wsdl:output>
        </wsdl:operation>

```

```
<wsdl:operation name="assignId">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="assignIdRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://wsv2.escholar.com" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="assignIdResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="createNewId">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="createNewIdRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://wsv2.escholar.com" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="createNewIdResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="auditResolveNearMatchEntry">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="auditResolveNearMatchEntryRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://wsv2.escholar.com" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="auditResolveNearMatchEntryResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="NearMatchResolutionV2Service">
  <wsdl:port binding="impl:NearMatchResolutionV2SoapBinding"
name="NearMatchResolutionV2">
    <wsdlsoap:address
location="https://tea4avdplbprod.tea.state.tx.us/uid/services/NearMatchResolutionV2"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

StudentSearch WSDL, operation search
StudentSearch WSDL, operations: searchStudent and searchWithIDExactMatch

```
<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="http://localhost:8080/uid/services/StudentSearch"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/uid/services/StudentSearch"
  xmlns:intf="http://localhost:8080/uid/services/StudentSearch"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="http://www.escholar.com"
  xmlns:tns2="http://util.java" xmlns:tns3="http://sql.java" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema targetNamespace="http://localhost:8080/uid/services/StudentSearch"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://sql.java" />
      <import namespace="http://util.java" />
      <import namespace="http://www.escholar.com" />
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="ArrayOf_xsd_string">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]" />
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://www.escholar.com" xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://sql.java" />
      <import namespace="http://util.java" />
      <import namespace="http://localhost:8080/uid/services/StudentSearch" />
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="StudentV2">
        <sequence>
          <element name="SSN" nillable="true" type="xsd:string" />
          <element name="alternateId" nillable="true" type="xsd:string" />
          <element name="alternateSource" nillable="true" type="xsd:string" />
          <element name="birthDate" nillable="true" type="xsd:string" />
          <element name="cdfNumbersArray" nillable="true" type="impl:ArrayOf_xsd_string" />
          <element name="cdfValuesArray" nillable="true" type="impl:ArrayOf_xsd_string" />
          <element name="districtCode" nillable="true" type="xsd:string" />
          <element name="ethnicityIndicator" nillable="true" type="xsd:string" />
          <element name="gender" nillable="true" type="xsd:string" />
          <element name="gradeLevel" nillable="true" type="xsd:string" />
          <element name="localStudentId" nillable="true" type="xsd:string" />
          <element name="multiracialBinarySum" nillable="true" type="xsd:double" />
          <element name="previousLastName" nillable="true" type="xsd:string" />
          <element name="race2Code" nillable="true" type="xsd:string" />
          <element name="race3Code" nillable="true" type="xsd:string" />
          <element name="race4Code" nillable="true" type="xsd:string" />
          <element name="race5Code" nillable="true" type="xsd:string" />
          <element name="raceEthnicCode" nillable="true" type="xsd:string" />
          <element name="recordUpdateDate" nillable="true" type="xsd:string" />
          <element name="recordUpdateDt" nillable="true" type="xsd:dateTime" />
          <element name="residentDistrictCode" nillable="true" type="xsd:string" />
          <element name="schoolCode" nillable="true" type="xsd:string" />
          <element name="schoolYear" nillable="true" type="xsd:string" />
          <element name="sourceSISCode" nillable="true" type="xsd:string" />
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>
</wsdl:definitions>
```

```

<element name="stateStudentId" nillable="true" type="xsd:string" />
<element name="studentFirstName" nillable="true" type="xsd:string" />
<element name="studentLastName" nillable="true" type="xsd:string" />
<element name="studentMiddleName" nillable="true" type="xsd:string" />
<element name="studentSuffix" nillable="true" type="xsd:string" />
  </sequence>
</complexType>
<complexType name="PotentialMatchV2">
  <sequence>
<element name="aliasId" nillable="true" type="xsd:string" />
<element name="alternateSource" nillable="true" type="xsd:string" />
<element name="cdfNumbersArray" nillable="true" type="impl:ArrayOf_xsd_string" />
<element name="cdfRefNumber" nillable="true" type="xsd:decimal" />
<element name="cdfValuesArray" nillable="true" type="impl:ArrayOf_xsd_string" />
<element name="claimRequired" type="xsd:boolean" />
<element name="comments" nillable="true" type="xsd:string" />
<element name="creationDate" nillable="true" type="xsd:anyType" />
<element name="districtCode" nillable="true" type="xsd:string" />
<element name="dob" nillable="true" type="xsd:dateTime" />
<element name="ethnicCode" nillable="true" type="xsd:string" />
<element name="ethnicCodeDesc" nillable="true" type="xsd:string" />
<element name="ethnicityIndicator" nillable="true" type="xsd:string" />
<element name="ethnicityIndicatorDesc" nillable="true" type="xsd:string" />
<element name="firstName" nillable="true" type="xsd:string" />
<element name="gender" nillable="true" type="xsd:string" />
<element name="gradeLevel" nillable="true" type="xsd:string" />
<element name="lastName" nillable="true" type="xsd:string" />
<element name="matchProbability" type="xsd:double" />
<element name="middleName" nillable="true" type="xsd:string" />
<element name="multiracialBinarySum" nillable="true" type="xsd:double" />
<element name="previousLastName" nillable="true" type="xsd:string" />
<element name="race2Code" nillable="true" type="xsd:string" />
<element name="race2CodeDesc" nillable="true" type="xsd:string" />
<element name="race3Code" nillable="true" type="xsd:string" />
<element name="race3CodeDesc" nillable="true" type="xsd:string" />
<element name="race4Code" nillable="true" type="xsd:string" />
<element name="race4CodeDesc" nillable="true" type="xsd:string" />
<element name="race5Code" nillable="true" type="xsd:string" />
<element name="race5CodeDesc" nillable="true" type="xsd:string" />
<element name="relationType" nillable="true" type="xsd:string" />
<element name="schoolCode" nillable="true" type="xsd:string" />
<element name="schoolYear" nillable="true" type="xsd:string" />
<element name="sourceSISCode" nillable="true" type="xsd:string" />
<element name="ssn" nillable="true" type="xsd:string" />
<element name="stateStudentID" nillable="true" type="xsd:string" />
<element name="suffixName" nillable="true" type="xsd:string" />
  </sequence>
</complexType>
<complexType name="NearMatchArray">
  <complexContent>
<restriction base="soapenc:Array">
<attribute ref="soapenc:arrayType" wsdl:arrayType="tns1:PotentialMatchV2[]" />
  </restriction>
</complexContent>
</complexType>
<complexType name="StudentSearchResult">
  <sequence>
<element name="aliasId" nillable="true" type="xsd:string" />

```

```
<element name="batchNumber" nillable="true" type="xsd:string" />
<element name="cdfMap" nillable="true" type="xsd:anyType" />
<element name="fieldsWithErrors" nillable="true" type="impl:ArrayOf_xsd_string" />
<element name="potentialNearMatches" nillable="true" type="tns1:NearMatchArray" />
<element name="stateStudentId" nillable="true" type="xsd:string" />
<element name="status" nillable="true" type="xsd:string" />
<element name="submissionId" nillable="true" type="xsd:string" />
  </sequence>
</complexType>
</schema>
</wsdl:types>
<wsdl:message name="searchStudentResponse">
<wsdl:part name="searchStudentReturn" type="tns1:StudentSearchResult" />
  </wsdl:message>
<wsdl:message name="searchWithIDExactMatchRequest">
<wsdl:part name="student" type="tns1:StudentV2" />
<wsdl:part name="userId" type="xsd:string" />
<wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
<wsdl:message name="i">
<wsdl:part name="student" type="tns1:StudentV2" />
<wsdl:part name="userId" type="xsd:string" />
<wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
<wsdl:message name="searchWithIDExactMatchResponse">
<wsdl:part name="searchWithIDExactMatchReturn" type="tns1:StudentSearchResult" />
  </wsdl:message>
<wsdl:portType name="StudentSearch">
<wsdl:operation name="searchWithIDExactMatch" parameterOrder="student userId password">
<wsdl:input message="impl:searchWithIDExactMatchRequest" name="searchWithIDExactMatchRequest" />
<wsdl:output message="impl:searchWithIDExactMatchResponse" name="searchWithIDExactMatchResponse" />
  </wsdl:operation>
<wsdl:operation name="searchStudent" parameterOrder="student userId password">
<wsdl:input message="impl:searchStudentRequest" name="searchStudentRequest" />
<wsdl:output message="impl:searchStudentResponse" name="searchStudentResponse" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="StudentSearchSoapBinding" type="impl:StudentSearch">
<wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="searchWithIDExactMatch">
<wsdlsoap:operation soapAction="" />
<wsdl:input name="searchWithIDExactMatchRequest">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  namespace="http://wsv2.escholar.com" use="encoded" />
  </wsdl:input>
<wsdl:output name="searchWithIDExactMatchResponse">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  namespace="http://localhost:8080/uid/services/StudentSearch" use="encoded" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="searchStudent">
<wsdlsoap:operation soapAction="" />
<wsdl:input name="searchStudentRequest">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  namespace="http://wsv2.escholar.com" use="encoded" />
  </wsdl:input>
<wsdl:output name="searchStudentResponse">
```

```
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"  
  namespace="http://localhost:8080/uid/services/StudentSearch" use="encoded" />  
</wsdl:output>  
</wsdl:operation>  
</wsdl:binding>  
<wsdl:service name="StudentSearchService">  
<wsdl:port binding="impl:StudentSearchSoapBinding" name="StudentSearch">  
<wsdlsoap:address location="http://localhost:8080/uid/services/StudentSearch" />  
</wsdl:port>  
</wsdl:service>  
</wsdl:definitions>
```

Appendix C – IDAssignmentV2 Code Basic Example (VB)

The code below is a basic example of the IDAssignmentV2 Web Service in Visual Basic.

```
Public Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    'create new IDAssignmentV2 web service
    Dim Service As DEMO.IDAssignmentService = New
DEMO.IDAssignmentService()
    'create new object for input and new result object
    Dim student As DEMO.Student = New DEMO.Student
    Dim StudentInformation As DEMO.IDAssignmentResultV2

    'set the student object from field values
    student.birthDate = DOB.Text
    student.studentFirstName = FirstName.Text
    student.studentLastName = LastName.Text
    student.gradeLevel = GradeLevel.Text
    student.districtCode = DistrictCode.Text
    student.raceEthnicCode = EthnicCode.Text
    student.schoolYear = SchoolYear.Text
    student.residentDistrictCode = ResidentDistrict.Text
    student.schoolCode = SchoolCode.Text
    student.gender = Gender.Text
    student.localStudentId = Localid.Text
    student.studentMiddleName = MiddleName.Text
    student.stateStudentId = StateID.Text

    'call the WS using the super user account username and password
    StudentInformation = Service.assignId(student, "abcd", "abcd")

    'write out details
    Response.Write("Batch Num:" & StudentInformation.batchNumber)
    Response.Write("<BR>")
    Response.Write(StudentInformation.status)
    Response.Write("<BR>Submission ID: ")
    Response.Write(StudentInformation.submissionId)
    Response.Write("<BR>State ID: ")
    Response.Write(StudentInformation.stateStudentId)
    Response.Write("<BR>")

    'write out all of the potential matches if they exists
    If Not (StudentInformation.potentialNearMatches()) Is Nothing Then
        Dim x As Long
        For x = LBound(StudentInformation.potentialNearMatches()) To
            UBound(StudentInformation.potentialNearMatches())
            Response.Write("Claims: ")
            Response.Write(StudentInformation.potentialNearMatches(x).claimRequired())
            Response.Write("<BR>Comments: ")
            Response.Write(StudentInformation.potentialNearMatches(x).comments())
            Response.Write("<BR>Create Date: ")
            Response.Write(StudentInformation.potentialNearMatches(x).creationDate)
            Response.Write("<BR>District Code: ")
            Response.Write(StudentInformation.potentialNearMatches(x).districtCode())
            Response.Write("<BR>Date of Birth: ")
            Response.Write(StudentInformation.potentialNearMatches(x).dob())
            Response.Write("<BR>Ethnic Code: ")
            Response.Write(StudentInformation.potentialNearMatches(x).ethnicCode())
            Response.Write("<BR>First Name: ")
            Response.Write(StudentInformation.potentialNearMatches(x).firstName())
            Response.Write("<BR>Gender: ")
            Response.Write(StudentInformation.potentialNearMatches(x).gender())
            Response.Write("<BR>Grade Level: ")
```

```
End If

Response.Write(StudentInformation.potentialNearMatches(x).gradeLevel())
Response.Write("<BR>Last Name: ")
Response.Write(StudentInformation.potentialNearMatches(x).lastName())
Response.Write("<BR>Match Probability: ")
Response.Write(StudentInformation.potentialNearMatches(x).matchProbability())
Response.Write("<BR>Middle Name: ")
Response.Write(StudentInformation.potentialNearMatches(x).middleName())
Response.Write("<BR>School Code: ")
Response.Write(StudentInformation.potentialNearMatches(x).schoolCode())
Response.Write("<BR>School Year: ")
Response.Write(StudentInformation.potentialNearMatches(x).schoolYear())
Response.Write("<BR>SSN: ")
Response.Write(StudentInformation.potentialNearMatches(x).ssn())
Response.Write("<BR>State Student ID: ")
Response.Write(StudentInformation.potentialNearMatches(x).stateStudentID())
Response.Write("<BR>Suffix: ")
Response.Write(StudentInformation.potentialNearMatches(x).suffixName()) Next

'write out individual fields with errors if validation
error exists If Not (StudentInformation.fieldsWithErrors())
Is Nothing Then
    Response.Write("Error(s):
    <BR> ") Dim i As Long

End If End Sub

'write out all of the fields
For i = LBound(StudentInformation.fieldsWithErrors()) To
    UBound(StudentInformation.fieldsWithErrors())
    Response.Write(StudentInformation.fieldsWithErrors(i)) Response.Write("<BR>")
Next
```