NIAS

NSIP Integration As a Service

NAPLAN Student Registration Data Validation

NAPLAN Results Reporting Data - NAPRR

Installation and User Guide v0.4

Version history

Version	Date	Notes
V01	4/07/2017	Initial release
V02	10/07/2017	Added Mac user install information
V03	15/08/2017	Added graphql information
V04	09/10/2017	Added v0.99 updates

Contents

Section 1 - NSIP Integration As a Service (NIAS)

1.1 What is NIAS?

NIAS is a suite of open-source components designed to enable as many different users as possible to quickly and easily solve issues of system integration using the Australian SIF Data Model for education. While NIAS includes generic functionality for system integration around SIF, the use of NIAS documented here is for the processing of data around NAPLAN.

The components in the current release perform the following functions:

- Validation of Student Registration data for NAPLAN
- Reporting of NAPLAN Results and Reporting data
- Facility for user generated queries on Results and Reporting data using GraphQL
- Audit validation between Student Registration data and Results and Reporting data

These tools are provided for the use of Test Administration Authorities (TAA) and jurisdictions responsible for the upload of NAPLAN Online student registration data to the National Assessment Platform, and the download and processing of NAPLAN results and reporting datasets from the Platform.

NIAS Data Validation (NAPVAL)

The data validation tool allows student registration data files in either .csv or .xml format to be validated to check data format, that mandatory fields in the files are populated, and that the fields are valid against the Registration Data Set specifications.

NAPVAL will also convert .csv files to .xml SIF format once the user is satisfied with the validation.

The user loads the student registration file into the interface, and the validation tool will produce a report of any errors or warnings found, which can be viewed on screen or downloaded. Once the file is validated, users can be confident in uploading the file to the National Assessment Platform for student registration.

NAPLAN Results and Reporting (NAPRRQL)

NAPRRQL is a package allowing reporting and browsing of NAPLAN results and reporting data files, for a particular state, sector or schools, provided as a SIF .xml file from the National Assessment Platform.

The user loads the results and reporting file, and NAPRRQL will produce a number of reports for a selected school which can be viewed on screen. The reports interface allow drill down of row data, as well as generation of .csv reports. The package can also be used on the command line to generate .csv files for each schools included and for the entire file.

There is also a facility for detailed examination of results data using GraphQL, allowing users to execute their own queries on the reporting data included in the file.

Audit Differences (NAPCOMP)

NAPCOMP allows comparison between a NAPLAN Student Registration file (in csv format) and a corresponding Results and Reporting data file received for the same cohort.

The user loads both files into various folders and the executable produces a .txt file which calls out differences where students appear in only one file and not the other. For example a student may be in the registration data file, but not the results and reporting dataset or vice versa.

None of these components have dependencies and can be run independently at any time.

Please note that instructions and screenshots included in this document are created using sample data which may not reflect realistic scores or results.

1.2 Installation

Installation of the NIAS tool loads all of the components as described on previous pages.

Pre-requisites

NIAS can be installed on Windows (32 or 64-bit), Linux (32 or 64-bit), or Macintosh. For the Windows installation you will need a Windows PC with the latest version of either Google Chrome or Mozilla Firefox installed. The Macintosh version works on Chrome, Firefox and Safari.

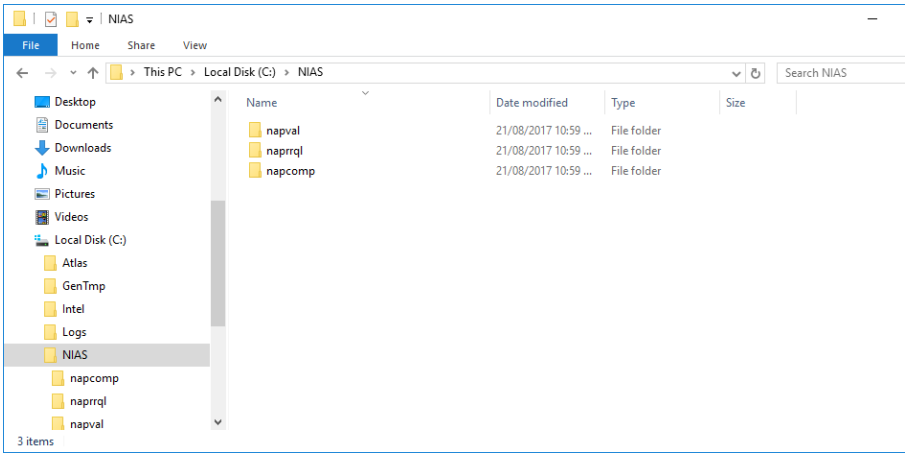
The NIAS toolset takes up approximately 30MB in hard disk space.

Installing files

1. Click on the following URL, or enter it in your browser:
   https://github.com/nsip/nias2/releases

2. Click on the relevant installation file to download the NIAS tools.

3. **Extract the contents of the go-nias.zip file** to a suitable high-level folder
   e.g.: C:\NIAS



Note: *Don't nest the folder structure too deeply in Windows; this can have an impact on permissible file name length.*

The installation loads 3 folders.

- **Napval** – enables validation of .csv or .xml files prior to loading into the NAPLAN Online Student Registration Management system
- **Naprrql** – converts results and reporting dataset into a number of NAPLAN school reports, provides tools for querying results and reporting data, and also contains an audit function
- **Napcomp**—contains an audit function allowing comparison of a NAPLAN Student Registration Data file against a NAPLAN Results and Reporting file.

Also included in the installation are sample files with which to run the NAPLAN Validation and the NAPLAN Results and Reporting tool. These are further discussed in the NAPVAL and NAPRRQL section of this document.

1.3 Updating & Removing NIAS Tools

The tools can be updated by deleting existing folders and downloading a newer version from *https://github.com/nsip/nias2/releases* (Subscribe to this link to be notified of updates to the NIAS tools.)

1.4 Conditions of Data Download

Education Services Australia Limited (ESA):

- complies with the Privacy Act 1988 (Cth) to maintain the privacy of personal information contained within each data extract or report including School Student Summary Report (SSSR) and Individual Student Report (ISR) (Data) while it is stored on the Assessment platform;
- is unable to control the use of the Data once it has been downloaded from the Assessment platform.

In order to maintain the privacy of the Data after you have downloaded it, prior to accessing and downloading the Data, you confirm and agree that:

- you are authorised to access and download the Data;
- your organisation has privacy and security controls in place in order to protect the privacy and security of the Data; and
- you will take all reasonable steps to ensure that the Data will not be misused, interfered with, lost, modified or disclosed to unauthorised personnel.

If you do not agree to the above conditions, **you must not** access and/or download the Data.

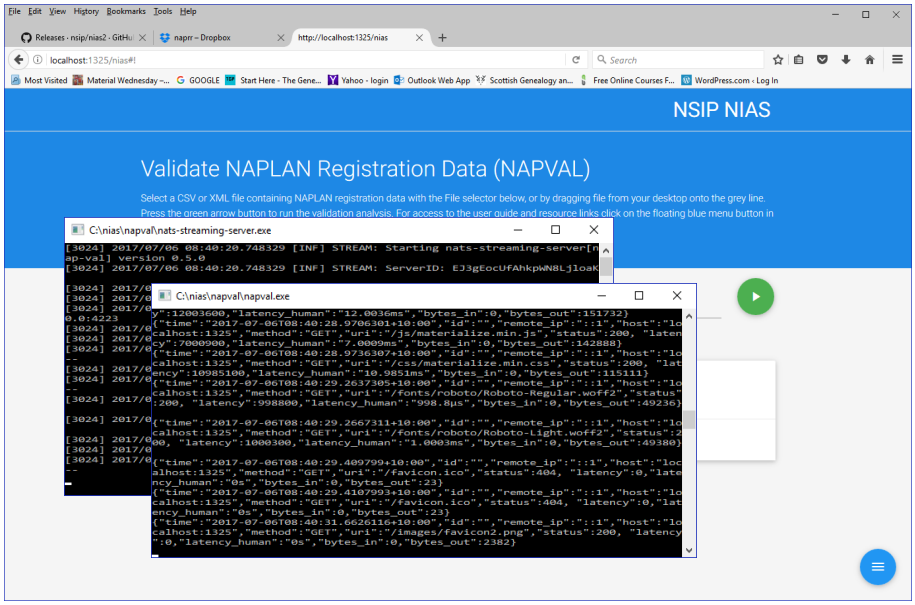
Section 2 - Validating NAPLAN Student Registration Data Using NAPVAL

Prior to commencing it is assumed that you have a NAPLAN Student Registration data file in the structure contained in the Student Registration Data Specifications v 0.95. The file can be .csv or .xml format. Data files in .csv format can be converted to .xml using this tool.

2.1 Running the Validation

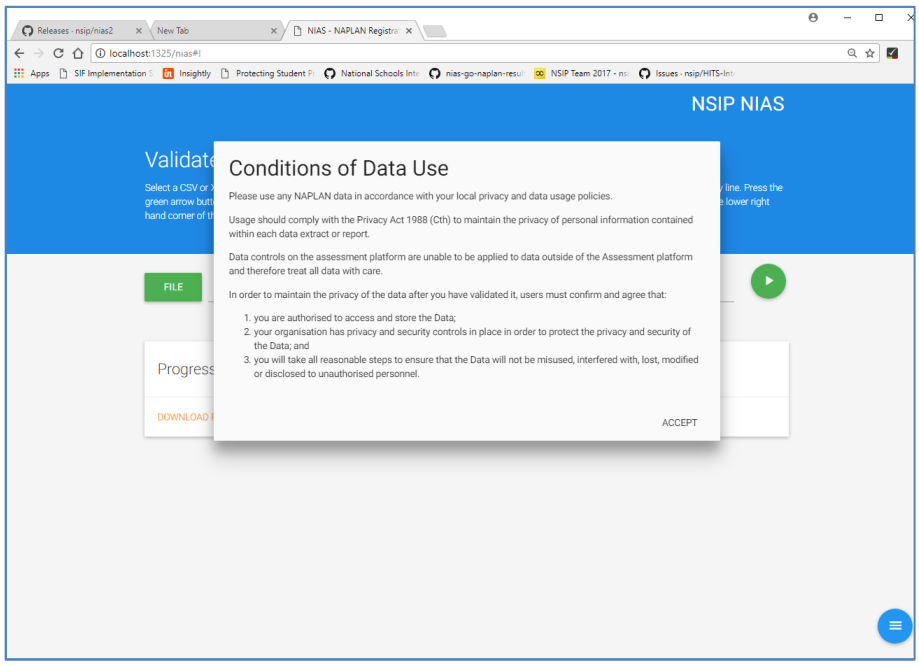
1. On Windows: To run the validation, navigate to the **napval** subfolder and double-click on **gonapval.bat**. This launches the various components and services of NIAS. On Macintosh and Linux: start **gonapval.sh** from the command line.
2. On launch in Windows, **gonapval.bat** opens a separate command prompt/terminal window for each of the NAPVAL key executables (napval.exe and nats-streaming-server.exe) and launches the NAPVAL web UI using your default web browser (Google Chrome and Mozilla Firefox are supported currently). On Macintosh and Linux, the other executables are launched in the same terminal window; you will need to access the browser to launch NAPVAL yourself, entering the address **localhost:1325**.

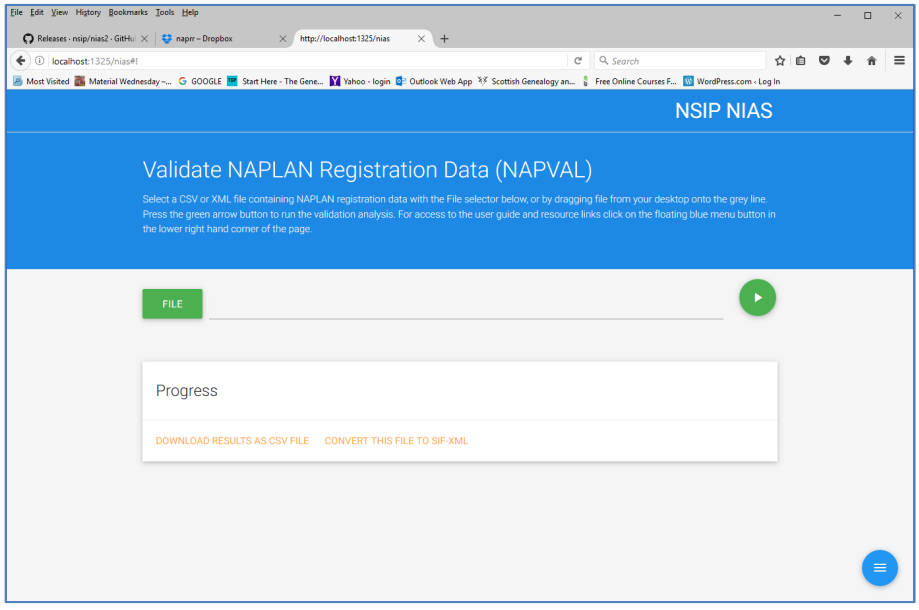
   Note: *If your default web browser is set to IE or another unsupported browser, the UI may not function correctly.*



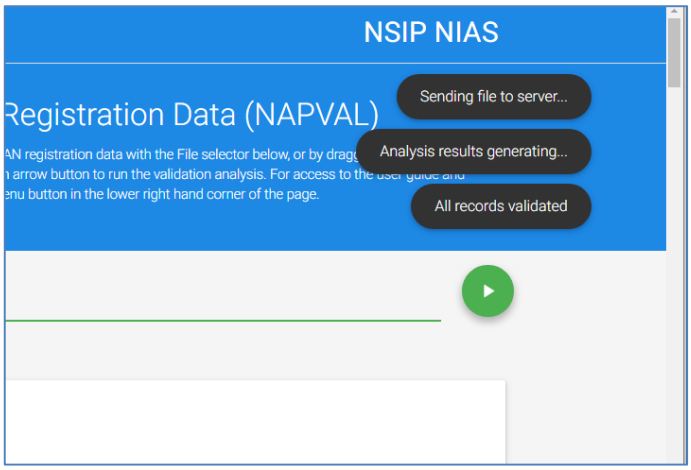
Important: These windows can be minimised but should not be closed whilst running NAPVAL. Closing these windows terminates that component of the NAPVAL validation tool.

3. The NAPVAL Web UI is displayed. Users are asked to confirm privacy policy on first run – click ACCEPT to continue.

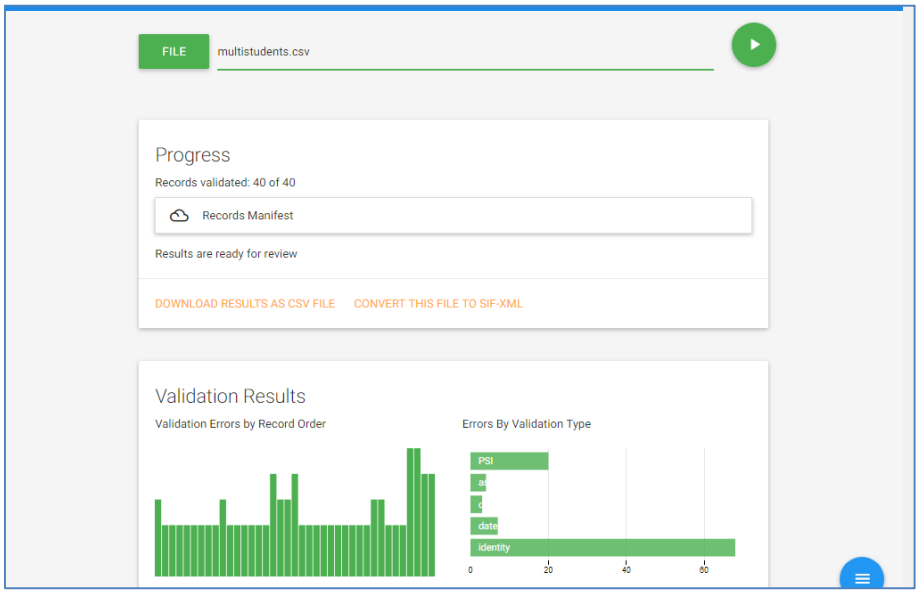




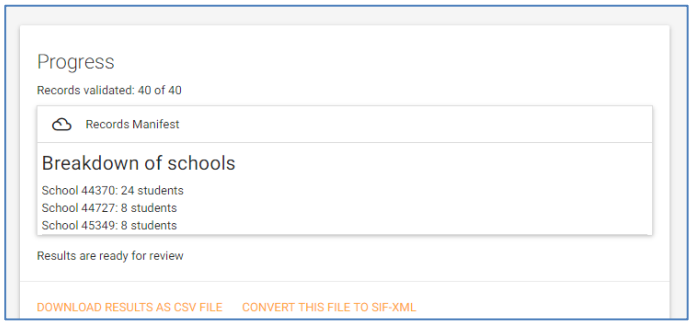
4. Select File and choose a file to be validated from the selection box and click Open, or drop the file in the input location. A sample file, students.csv, is included with the software distribution in the NAPVAL folder. NAPVAL validates files that are either CSV or XML file formats.
5. Click on the play button to commence validation.
6. Progress of the validation is displayed on screen. Do not click on the "Download Results as CSV File" link until all records are validated.



7. NAPVAL performs an initial validation, then lists a summary of the first 100 validation issues on screen.
8. Once file processing is complete, a list of detailed validation results is displayed on screen, broken up into three areas:
    - Validation errors by record order (graph on left)
    - Errors by validation type (graph on right)
    - Error details table – Errors ordered by original file line number (table at bottom)

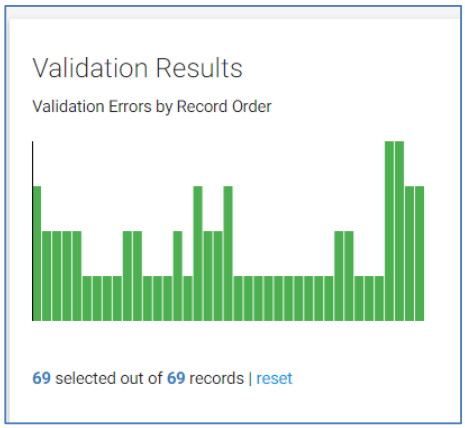


9. There is also a Records Manifest which when expanded will display the breakdown of students records per school. You can use the Manifest to confirm that the uploaded file contains the expected data.

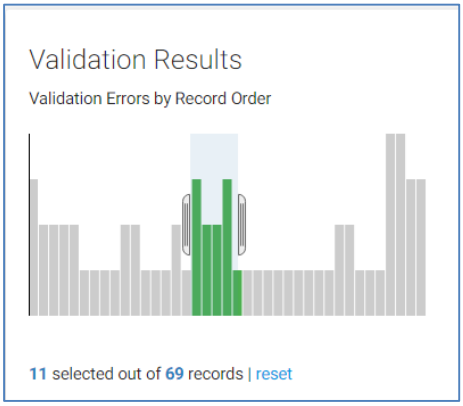


10. To view more detail on the validation errors, use one of the following controls:
    - Validation errors by record order (graph on left) or
    - Errors by validation type (graph on right)

10.1 **Validation errors by record order**

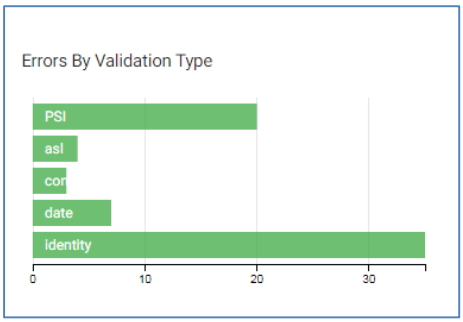


- This control allows a user the ability to narrow down the validation errors reported based on the record order.
- Move the mouse cursor over the column graph – a + symbol will appear. Drag this to select a section of the file (selected areas are green). Selecting a section will alter the errors by validation type data displayed and the details below in the errors ordered by original file line number table.
- As you move the selection window, the system updates the selection of errors displayed based on your selection.

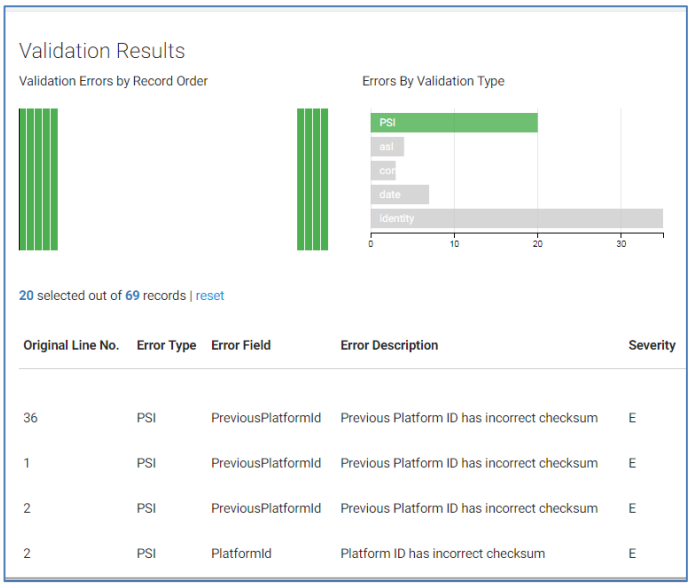


- To remove any filter on the errors reported, click on **reset.**

10.2 **Errors by validation type**

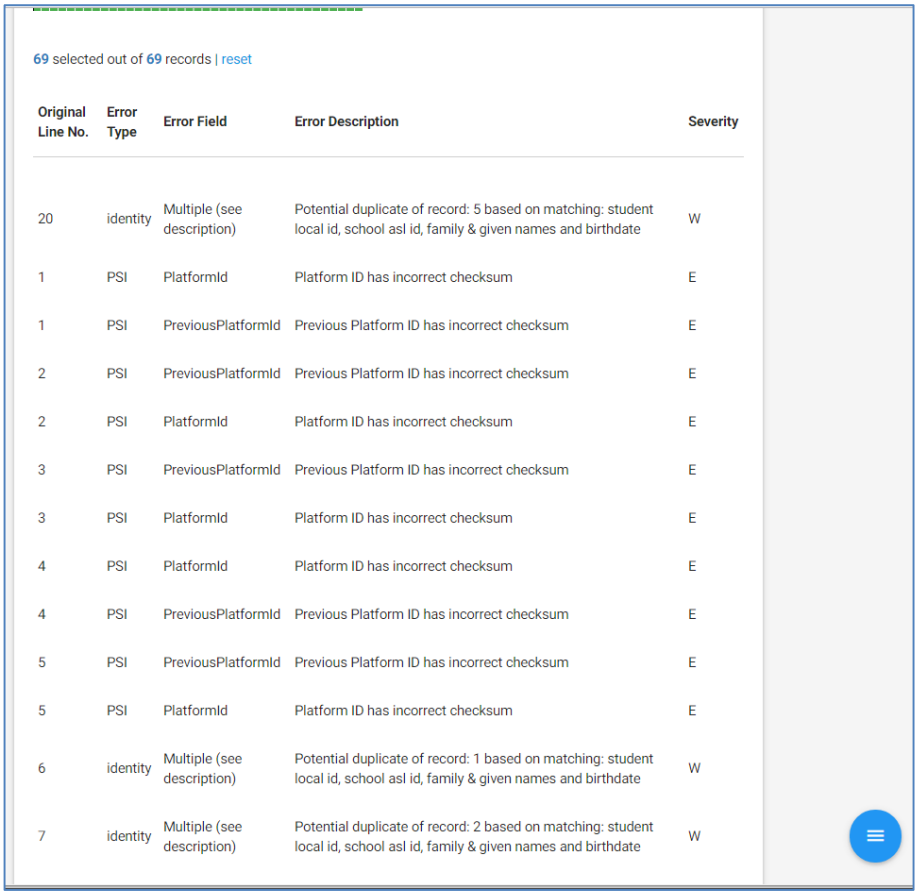


- This validation control allows the user to focus on a particular validation type (or types) by selecting the appropriate bar/s.
- For example, selecting PSI displays only the PSI related errors in the detail table and record order graphs.



- You can select more than one validation type to combine results (highlighted in green) by clicking on multiple bars in the graph.

10.3 **Error details table – Errors ordered by original file line number**



69 selected out of 69 records | reset

Original Line No.	Error Type	Error Field	Error Description	Severity
20	identity	Multiple (see description)	Potential duplicate of record: 5 based on matching: student local id, school asl id, family & given names and birthdate	W
1	PSI	PlatformId	Platform ID has incorrect checksum	E
1	PSI	PreviousPlatformId	Previous Platform ID has incorrect checksum	E
2	PSI	PreviousPlatformId	Previous Platform ID has incorrect checksum	E
2	PSI	PlatformId	Platform ID has incorrect checksum	E
3	PSI	PreviousPlatformId	Previous Platform ID has incorrect checksum	E
3	PSI	PlatformId	Platform ID has incorrect checksum	E
4	PSI	PlatformId	Platform ID has incorrect checksum	E
4	PSI	PreviousPlatformId	Previous Platform ID has incorrect checksum	E
5	PSI	PreviousPlatformId	Previous Platform ID has incorrect checksum	E
5	PSI	PlatformId	Platform ID has incorrect checksum	E
6	identity	Multiple (see description)	Potential duplicate of record: 1 based on matching: student local id, school asl id, family & given names and birthdate	W
7	identity	Multiple (see description)	Potential duplicate of record: 2 based on matching: student local id, school asl id, family & given names and birthdate	W

- The Error details table displays validation errors detected in the imported file (you can filter it using the control graphs mentioned above).
- The table describes the original record number, the type (ASL, content, identity), the field where the error was detected, the error description, and the error severity (Warning or Error).

11. To export the produced error report, select Download Results as CSV File. You will be able to open or save the file as appropriate. Error reports use the naming convention of OriginalFileName_error_report.csv.
12. After you have reviewed and corrected errors in the source files, you can re-load a new version of the file for review. It is recommended that you refresh the browser if the NAPVAL validation screen is still active. Go back to File, select the revised file, then repeat the file upload steps as detailed above.
13. To end NAPVAL processing:
    - On Windows, run **stopnapval.bat** (which will close each terminal window) and close the web browser, or
    - Close each of the terminal windows by clicking the **x** in the top right corner of each window and close the web browser.
    - On Macintosh and Linux, run **stopnapval.sh** and close the web browser.

2.2 NIAS Components

NIAS contains a number of validation tools. These validation services can be accessed by setting variables in the napval.toml configuration file. When you open this file in a text editor, it should look something like this:

```
# Baseline year for DOB checks
TestYear = "2016"

# Validations to invoke
# ValidationRoute = ["schema", "local", "id", "dob", "asl"]
ValidationRoute = ["schema", "id", "dob", "asl"]

# Data match fields for matching across schools
StudentMatch = ["FamilyName", "GivenName", "BirthDate"]

# Webserver port
WebServerPort = "1325"

# Number of validation engines
PoolSize = 4
```

Configurable variables

NIAS validation variable	Notes
TestYear	Sets the test year during which assessment is running. Used to ensure that students' birth dates align with their test levels.
ValidationRoute	Sets the validations to apply – see table below for details. Currently validations are not order-dependent.
StudentMatch	Fields of the student record to be used for data matching of records between schools during the *id* check.
WebServerPort	Sets the web server port. You can edit this if the default conflicts with another service.
NATSPort	Sets the port for the NATS streaming service (used as the bus for messages in the microservice architecture of the software). You can edit this if the default conflicts with another service.

Validation route values

NIAS validation variable	Notes
asl	Checks that ASL values are correct.
schema	Applies the NAPLAN registration data set validations defined in schemas/core.json.
schema2	Applies dependency validation on NAPLAN registration data set, as defined in schemas/core_parent2.json : ensuring that if one parent 2 value is provided, all of them are provided.
dob	Apply date of birth validation according to the setting of TestYear.
Id	Apply id validation: confirm that every student has a unique LocalId per school, a unique PSI per school, and a unique LocalId, Family Name, Given Name, and Birth Date per school. Any collisions within a school are reported. It also detects any students with the same StudentMatch fields between schools.
asl	Applies Australian School List validation: confirms that each ASL Id given for a school in the registration data exists in the Australian School List, and is the identifier for a school in the correct state (as given in the student record address).
psi	Apply PSI validation: verifies the check letter for each Platform Student Identifier in the file.
numericvalue	Validates all the numbers with value constraints in the file: currently this only applies to the FTE value, which the service confirms is a decimal number between 0 and 1.
local	Applies any validations you have set in schemas/local.json.

	By default this file is not processed, and the default local.json file shipped with nias performs no validation. If you wish to use local validation, rename your local validation file to local.json (more details below).

2.3 NIAS Validation Schemas

Validation schemas (in JSON format) are used by NIAS to validate that the input files contain the required mandatory fields and contain valid values. NIAS comes with two validation schemas located in the **schemas** folder and an optional third local validation:

1. Core (*core.json)* – This is the validation schema based on the approved registration data set. It provides the core validation for mandatory/optional fields and valid values. Core validation is always applied when validating file contents. **Do not modify this file.**
2. Core, Parent 2 (*core_parent2.json*) - This is the validation schema expressing the dependencies between Parent 2 demographic values in registration data: If one such field is present, all of them need to be present. **Do not modify this file.**
3. Local (*local.json)*- This is an optional extra layer of validation which can be applied in addition to the core validation. You can edit this file to include local validations as required.

local.json

The file local.json can supply an optional extra layer of validation in addition to the validation found in core.json. Typical examples of where extra local validation may be useful include:

1. Making additional fields mandatory (for example a jurisdiction wide identifier)
2. Modifying allowed values (for example removing year level 0)
3. Modifying min/max lengths (for example all jurisdiction identifiers must be 9 characters in length)

The file local.json contains examples of possible local validations which users may seek to implement including a max length of 10 for home group, max length of 36 for jurisdiction id, and an absolute length of 8 for TAAId. It also lists TAAId, JurisdictionId and HomeGroup as mandatory fields.

A suitable JSON or text editor is recommended.

Enabling local schemas

NIAS users may seek to apply one or more localised schemas (for example SouthAustralia.json, SACatholics.json, SAIndependents.json).

To enable your own additional validation schema:

1. In the **napval\schemas** folder, rename local.json to defaultlocal.json
2. Rename your local validation file to local.json
3. Place it in the **schemas** folder
4. In the napval.toml file, uncomment the ValidationRoute with “local” included in the variable arguments, and comment out the default ValidationRoute.
5. Restart NAPVAL and validate your file/s.

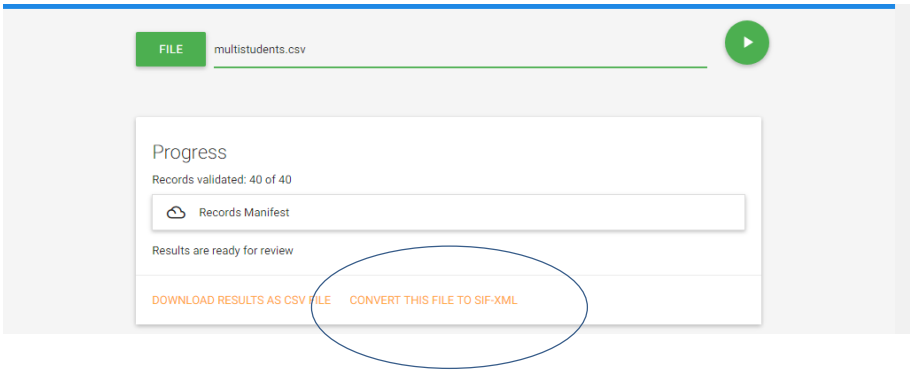
Alternatively, you can implement your own validations by directly editing the default local.json file.

2.4 Conversion of CSV to XML

Once you are happy with the level of validation, you can convert CSV files to SIF XML format.

1. In the NAPVAL tool select File.
2. Choose the relevant CSV file (this file would typically be the file that has been worked on and validated as required). Select Open.
3. Click the 'Convert this file to XML' link,

4. Save the XML file to an appropriate location. The file will have the same filename as the CSV file but with an XML file extension.
5. You can convert files as many times as you require.



Section 3 - Using NAPRRQL

NAPRRQL comes with a full results reporting dataset sample extract for 4 schools (master_nap.xml.zip), which is ingested on first running the application. In order to run your own data received from the National Assessment Platform, substitute this file with your own zipped XML file; NAPRRQL will read any file suffixed **.xml.zip** in the **\in** folder. Please ensure that your own zipped XML file is not password-protected; you may need to unzip the password-protected file received from the national assessment platform, and re-zip it without a password.

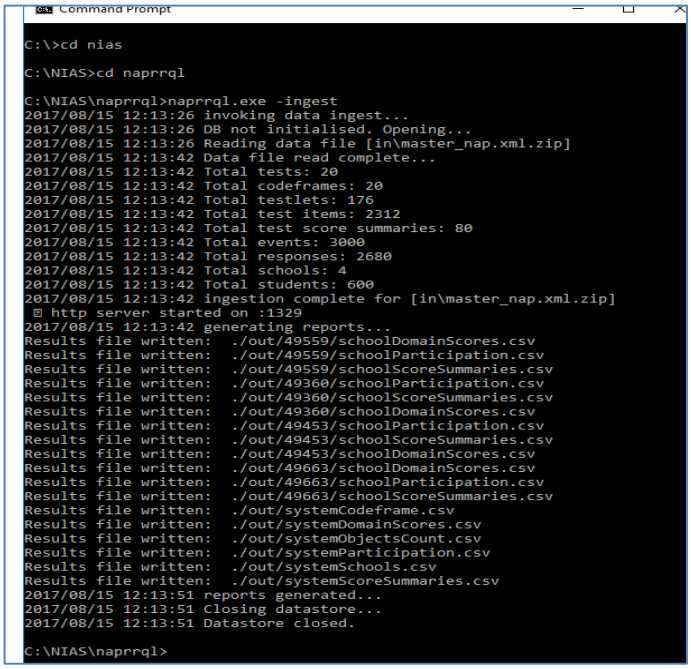
When the application is run, data is streamed from the data file in the **\in** folder and processed onto the reporting sub-system. Report generators are then run to produce .csv extracts of the data in the **\out** folder. This contains aggregate reports at the top level for all report types (score summaries, domain scores, participation and code-frame).

A folder for each school is created under the **\out** folder, with each subfolder named according to the ASL Id (ACARA Id) of the school. These folders contain the reports for each school.

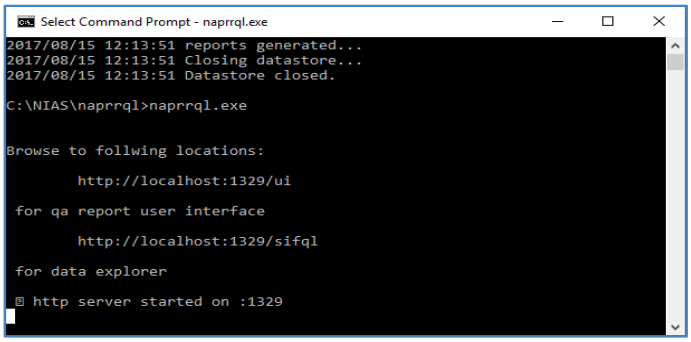
The code-frame report is only generated at the top level as it is about the test, not about an individual school.

3.1 NAPRRQL Sample data

1. To run NAPRRQL and populate it from the sample file, navigate to the command prompt and to the **naprrql** folder (eg. C:\nias\naprrql).
2. Run **naprrql.exe -ingest** on Windows, **naprrql -ingest** on Linux and Macintosh. This launches the various components and services of NAPRRQL and begins processing of any data files in the **\in** folder.
3. The application ingests the data in readiness for generating reports for each school, and allows viewing of the Results Reporting data in the web user interface in the next steps.
    - Participation Report
    - School Score Summary Report
    - Domain Score Report



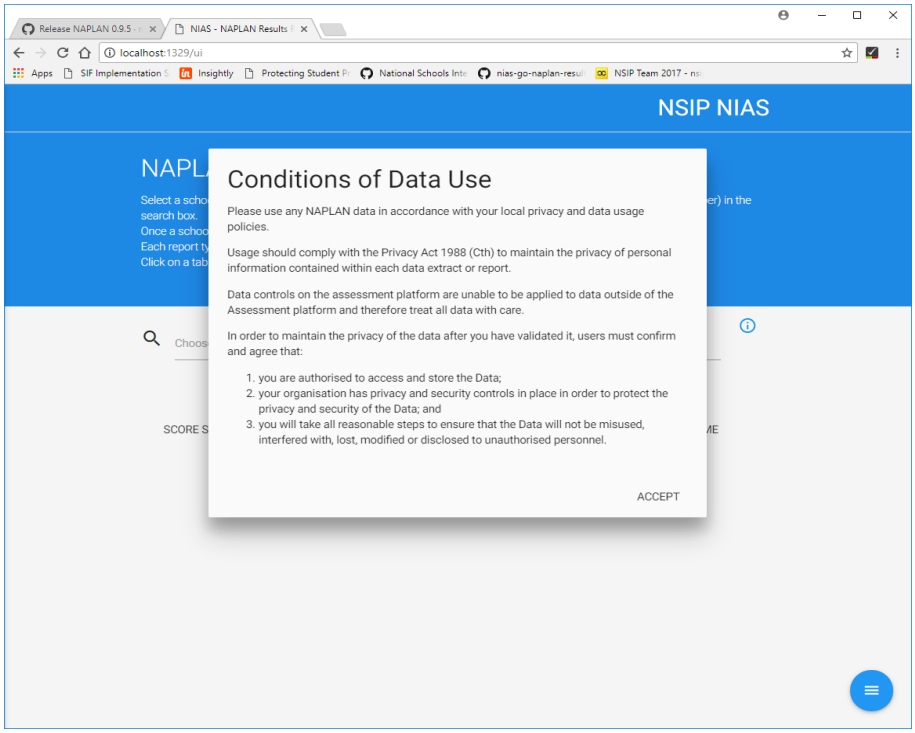
4. Once the report generation is complete, to run the web interface to allow browsing of NAPLAN Results Reporting run **naprrql.exe** on Windows, **naprrql** on Linux and Macintosh.



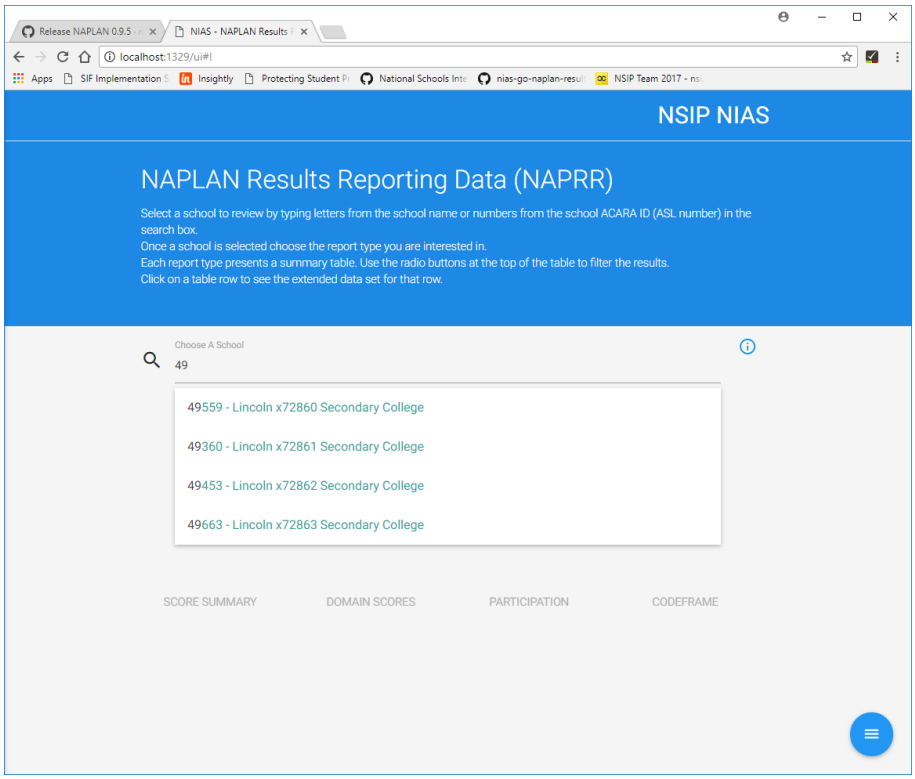
Important: The window can be minimised but should not be closed whilst running NAPRRQL. Closing the window terminates that component of the NAPRRQL reporting tool.

5. Open your browser (Google Chrome or Mozilla Firefox) and navigate to:
   http://localhost:1329/ui

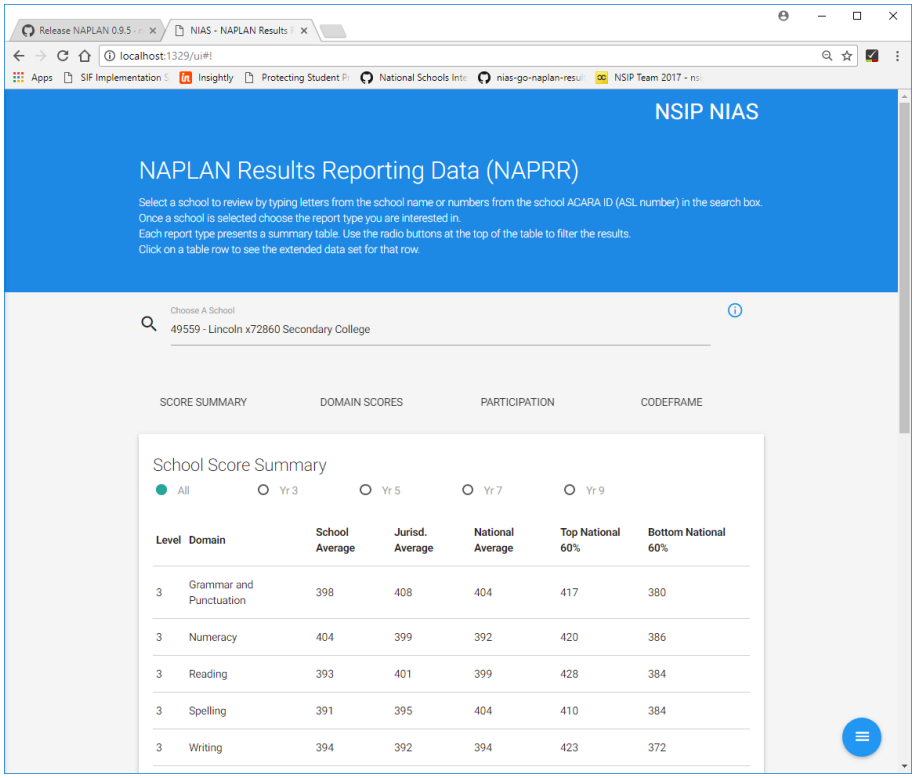
6. The NAPLAN Results Reporting Data Web User Interface will display. Click ACCEPT to agree to the Conditions of Data Use:



7. To begin viewing data, select a school from the search box by typing letters from the school name or the school ACARA ID and choosing the required school.

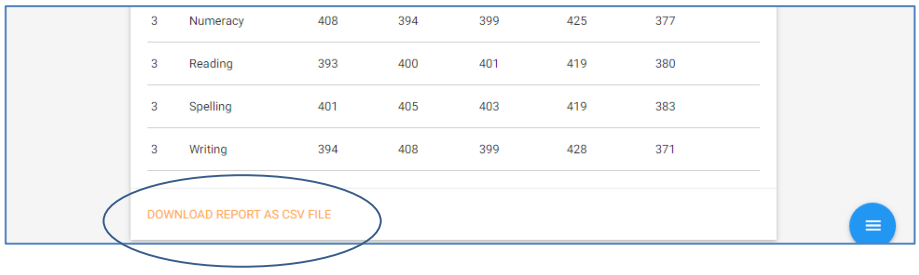


8. Once a school is selected its School Score Summary will be displayed. As the data is generated, the Domain Scores, Participation and Codeframe links will activate.

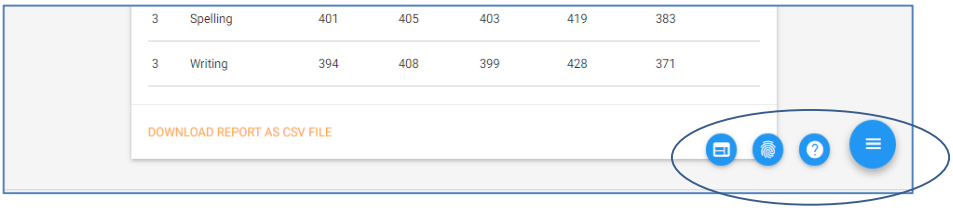


9. To navigate between reports available for the selected school, click on the links displayed below the school name.

10. Reports can be downloaded in csv format by clicking on the link at the bottom of the report.

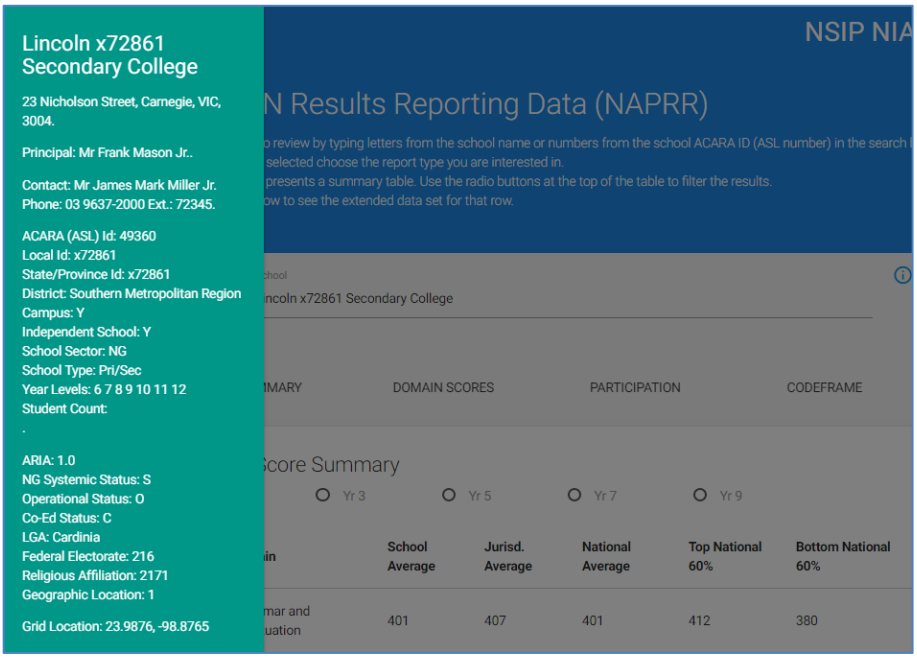


11. Clicking on the blue button at the bottom of the screen will open up menus for Help, Privacy and the Results Reporting Data Set Specification



12. To view information about the school displayed click on the  button to the right of the school name

13. Information will display on the left of the screen. Click back on the main screen to return to NAPRRQL reporting.



3.2 Loading data files and generating reports

Important Note: Because the installation for NAPRRQL includes sample files for testing, it is important that these files are removed first prior to copying the relevant data files to the folders as described below.

The NAPRRQL will process any .xml zip file it finds in the folder, which means it will process the sample file along with any files copied in. This also means that multiple files can be loaded into the **naprrql\in** folder and be processed at the one time.

The sample file that must be removed is the master_nap.xml.zip file contained in the folder: **\naprrql\in.**

Files and folders in the **\naprrql\out** folder will be deleted each time the NAPRRQL tool is run so it is not necessary to delete sample data from here.

Please refer back to section 3.1 NAPRRQL Sample data for screenshots if required.

1. To run NAPRRQL and populate it using data from the National Assessment Platform, ensure your data file is in .**xml** format and zipped
2. Copy the file to the **naprrql\in** folder.
3. Navigate to the installation folder in a console or terminal and run **naprrql.exe -ingest** on Windows, **naprrql -ingest** on Linux and Macintosh. This launches the various components and services of NAPRRQL and begins processing of data files in the **/in** folder.
4. The application ingests the data in preparation for report generation.
    - Participation Report
    - School Score Summary Report
    - Domain Score Report

    Note that the ingest process only needs to be done once unless the data changes.
5. Once the report generation is complete, to run the web interface for NAPLAN Results Reporting run **naprrql.exe** on Windows, **naprrql** on Linux and Macintosh
6. Processing the file can take a long time; the result set for a jurisdiction will likely take tens of minutes, and should be run on the fastest computer you have available.
7. **Important: The window can be minimised but should not be closed whilst running NAPRRQL. Closing the window terminates that component of the NAPRRQL reporting tool.**
8. Open your browser (Google Chrome or Mozilla Firefox) and navigate to: **http://localhost:1329/ui**
9. The NAPLAN Results Reporting Data Web User Interface will display.
10. To begin viewing data, select a school from the search box by typing letters from the school name or the school ACARA ID and choosing the required school.
11. Navigation of the available reports is explained in the following pages.

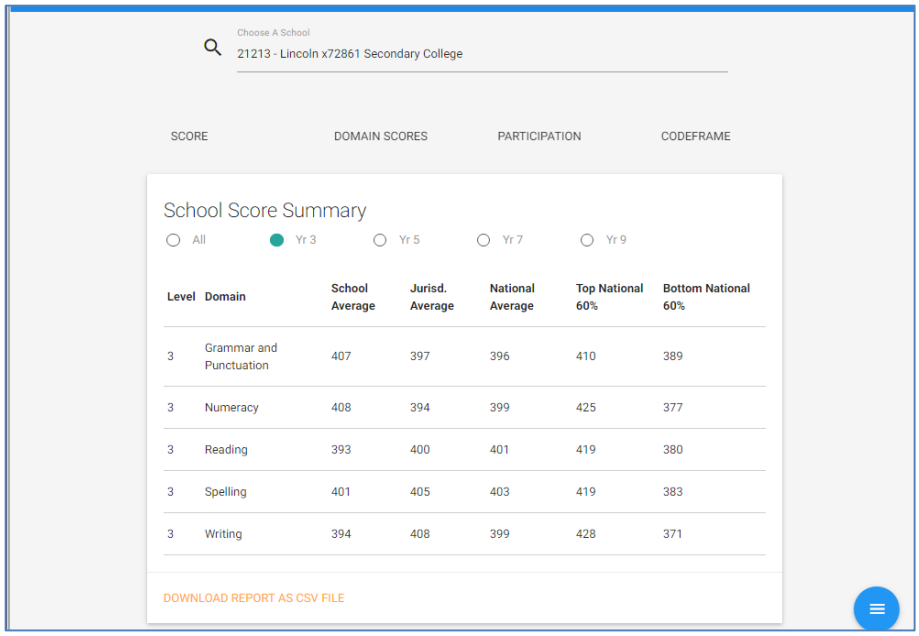
Note: Year 3 Writing data is currently included in the sample file; however the national assessment platform will not be used to deliver Year 3 Writing, so it will not be outputting Year 3 Writing results in SIF .xml format. Future functionality will enable Year 3 writing data from existing contractors to be included as an input to the NAPRRQL program, as a separate file in its native format.

3.3 Report Functionality

School Score Summary Report

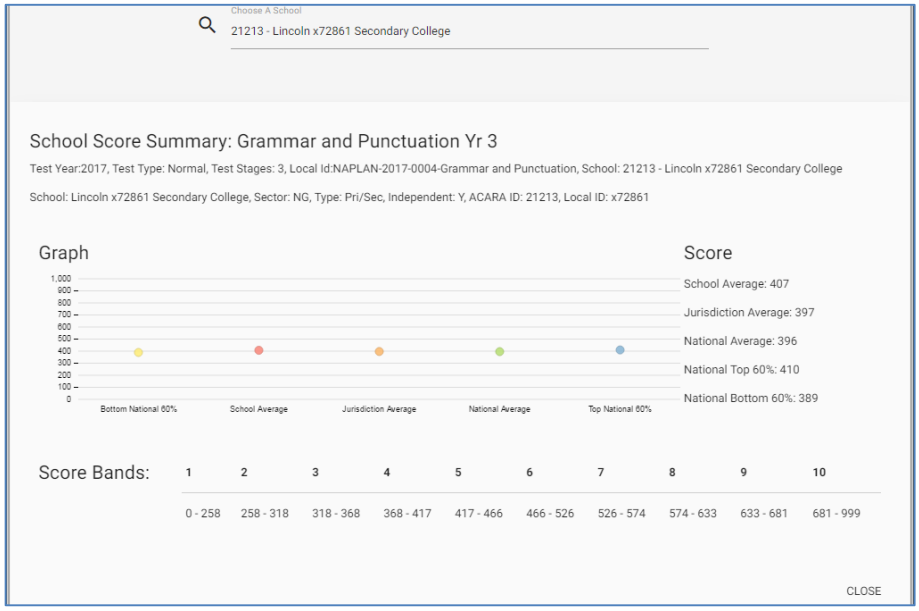
Select this report by clicking on the SCORE tab below the school name.

By default all domains for all year levels will be displayed. This can be filtered by Year Level by choosing the appropriate year level radio button at the top of the report.



The displayed report can be exported by clicking on the link at the bottom of the report

To view extended data for any particular row, click on the required row. A screen displaying further data graphically will be displayed. By hovering the mouse over the data points on the graph the values will display.

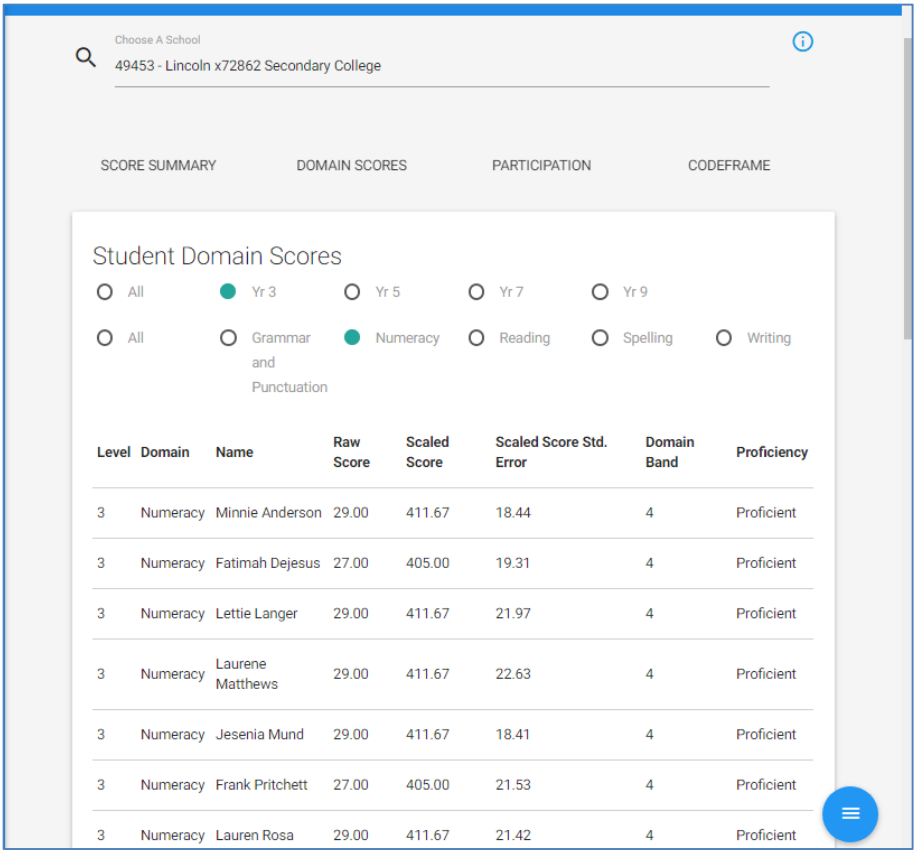


Click CLOSE to return to the previous screen.

Student Domain Scores Report

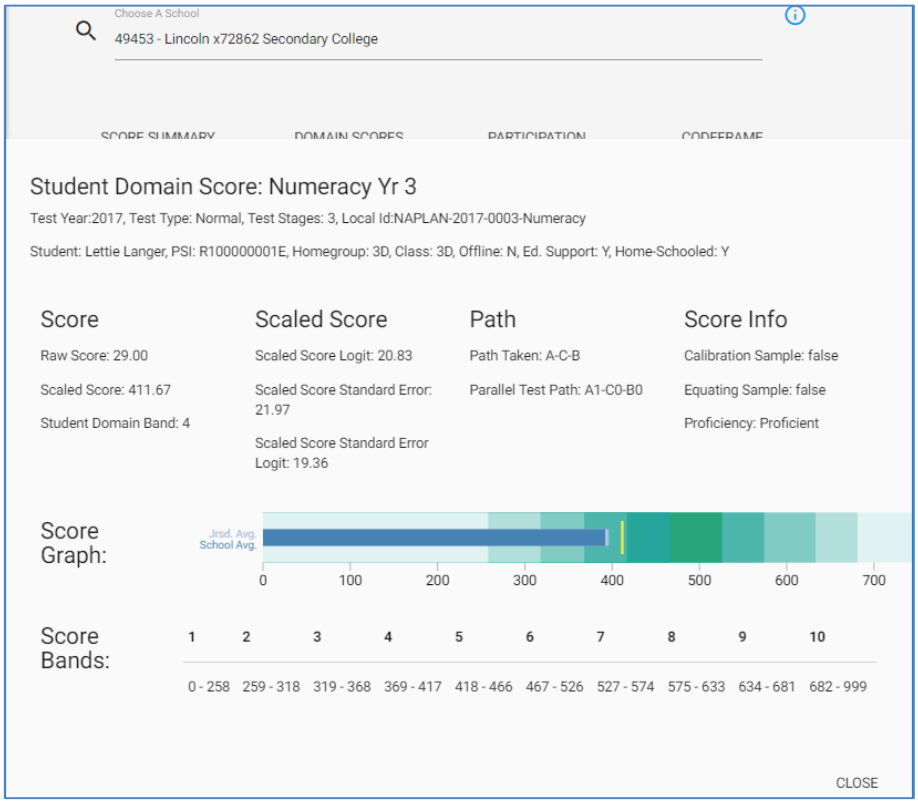
Select the required school. To access this report, click on the DOMAIN SCORES tab below the school name.

By default, all domains for all year levels will be displayed. This can be filtered by Year Level and/or Domain by choosing the appropriate radio buttons.



Level	Domain	Name	Raw Score	Scaled Score	Scaled Score Std. Error	Domain Band	Proficiency
3	Numeracy	Minnie Anderson	29.00	411.67	18.44	4	Proficient
3	Numeracy	Fatimah Dejesus	27.00	405.00	19.31	4	Proficient
3	Numeracy	Lettie Langer	29.00	411.67	21.97	4	Proficient
3	Numeracy	Laurene Matthews	29.00	411.67	22.63	4	Proficient
3	Numeracy	Jesenia Mund	29.00	411.67	18.41	4	Proficient
3	Numeracy	Frank Pritchett	27.00	405.00	21.53	4	Proficient
3	Numeracy	Lauren Rosa	29.00	411.67	21.42	4	Proficient

To view extended data for any particular row, click on the required row. A screen displaying further data graphically will be displayed.

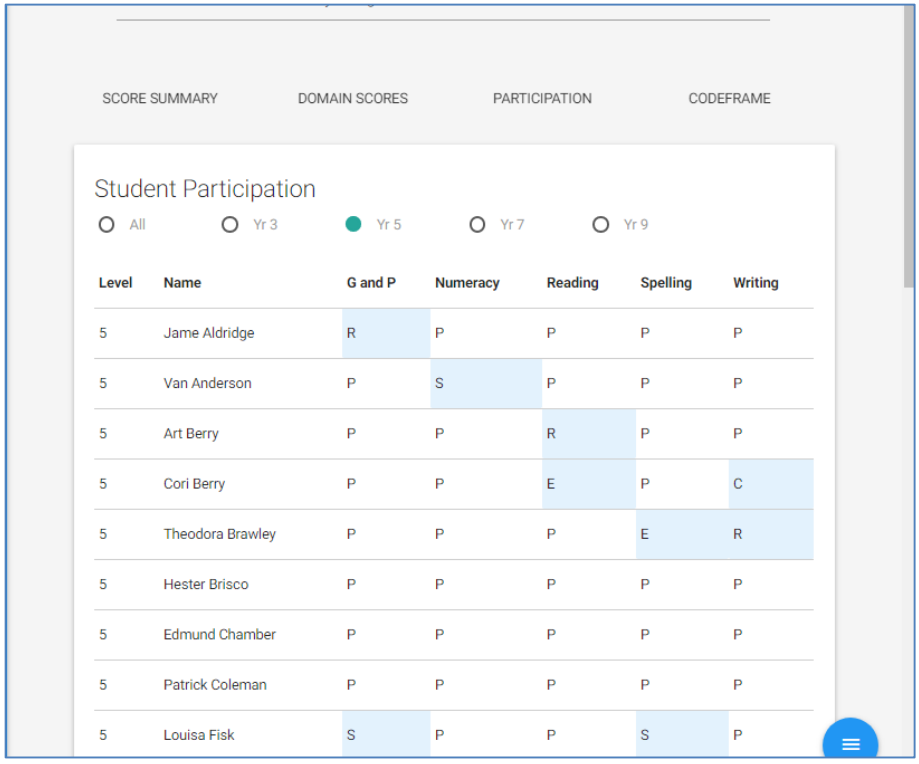


Click CLOSE to return to the previous screen.

Student Participation Report

Select the required school. To access this report, click on the PARTICIPATION tab below the school name.

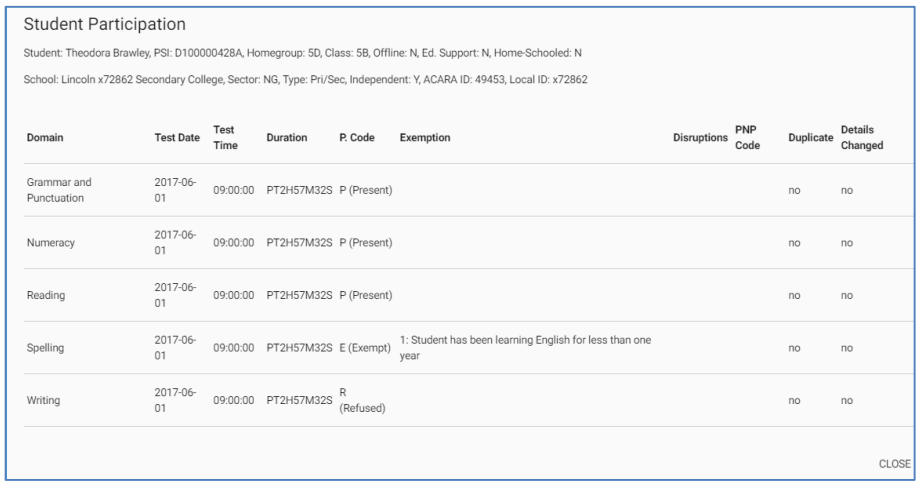
By default, all domains for all year levels will be displayed. This can be filtered by Year Level by choosing the appropriate radio buttons.



Level	Name	G and P	Numeracy	Reading	Spelling	Writing
5	Jame Aldridge	R	P	P	P	P
5	Van Anderson	P	S	P	P	P
5	Art Berry	P	P	R	P	P
5	Cori Berry	P	P	E	P	C
5	Theodora Brawley	P	P	P	E	R
5	Hester Brisco	P	P	P	P	P
5	Edmund Chamber	P	P	P	P	P
5	Patrick Coleman	P	P	P	P	P
5	Louisa Fisk	S	P	P	S	P

Participation codes other than P (present) are highlighted for easy identification on screen.

Click to view further data for a particular row. A pop up screen will display information about the student, the school, each test, participation codes and other events such as exemptions or disruptions.



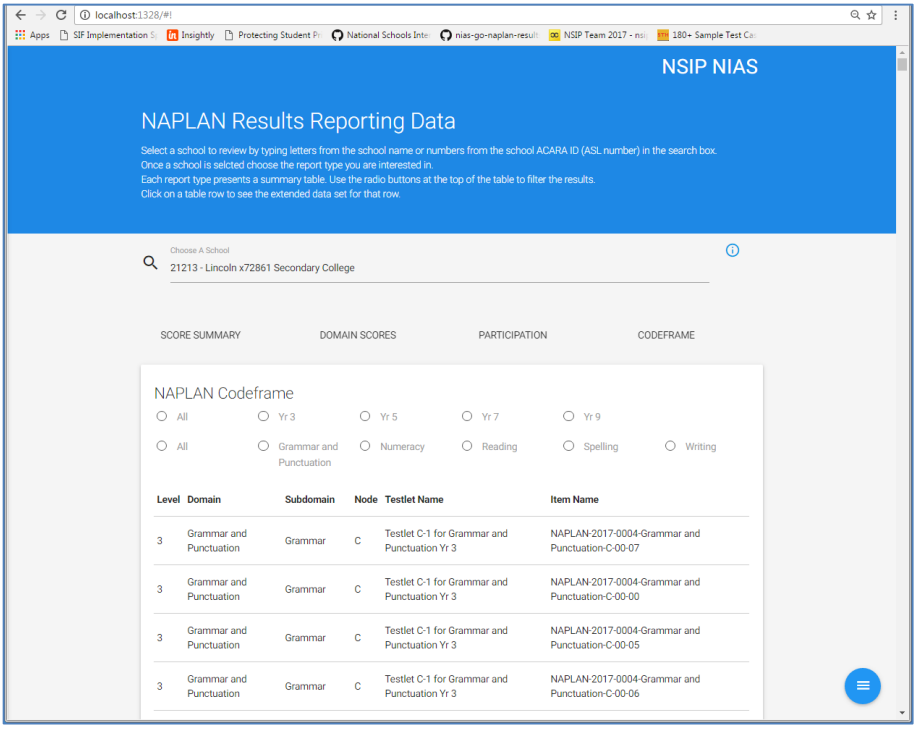
Domain	Test Date	Test Time	Duration	P. Code	Exemption	Disruptions	PNP Code	Duplicate	Details Changed
Grammar and Punctuation	2017-06-01	09:00:00	PT2H57M32S	P (Present)				no	no
Numeracy	2017-06-01	09:00:00	PT2H57M32S	P (Present)				no	no
Reading	2017-06-01	09:00:00	PT2H57M32S	P (Present)				no	no
Spelling	2017-06-01	09:00:00	PT2H57M32S	E (Exempt)	1: Student has been learning English for less than one year			no	no
Writing	2017-06-01	09:00:00	PT2H57M32S	R (Refused)				no	no

Click CLOSE to return to the previous screen.

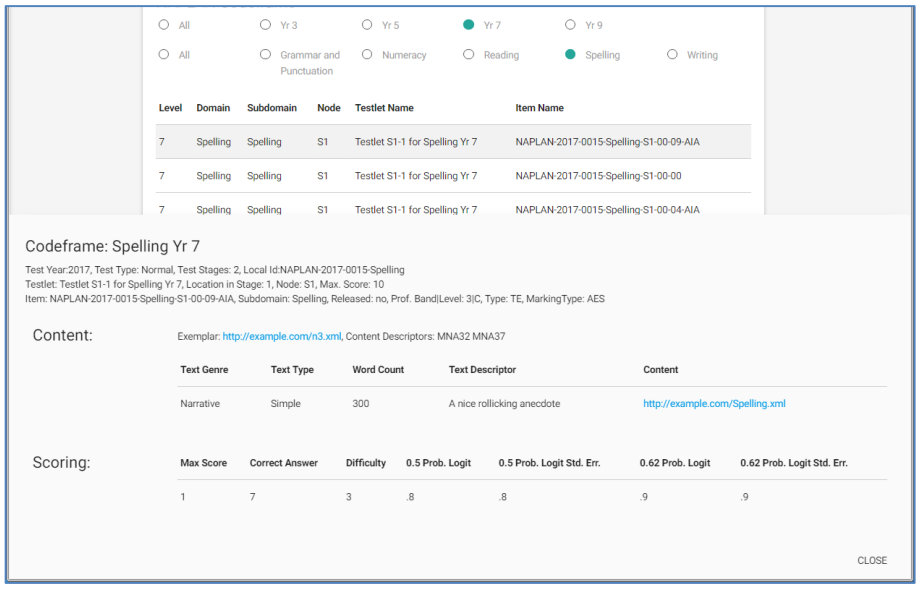
Codeframe Report

Select this report by clicking on the CODEFRAME tab below the school name.

By default, the NAPLAN Codeframe Report will display test items for all domains for all year levels. This can be filtered by Year Level and/or Domain by choosing the appropriate radio buttons at the top of the report.



Click to view further data for a particular row. A pop up screen displaying information about the test and testlet associated with the selected test item will display. Links to Exemplar and Content are also available on this page.



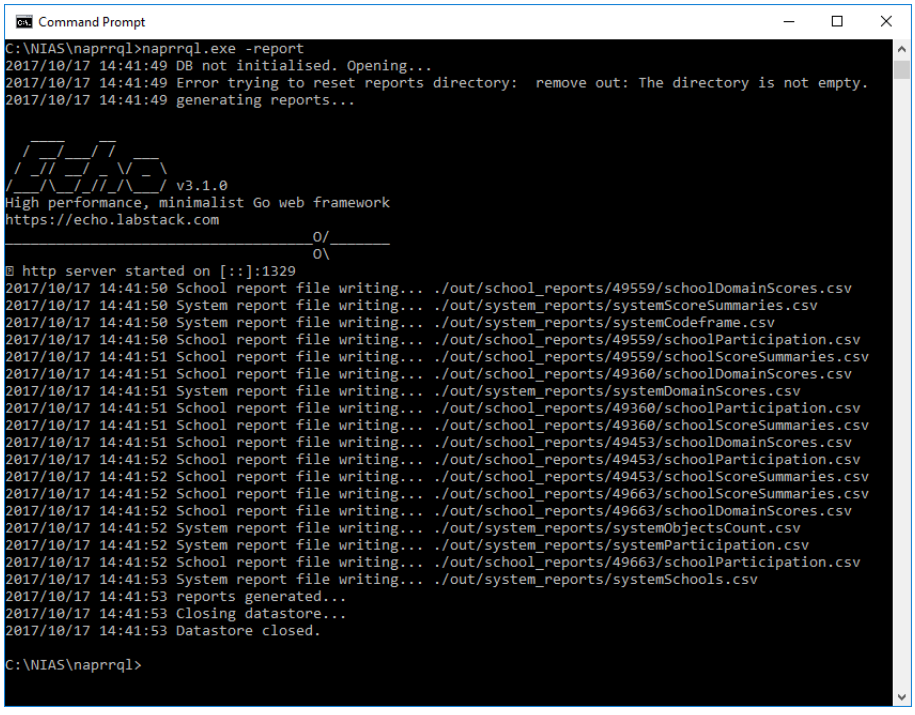
Click CLOSE to return to the previous screen.

3.4 Results Reporting

Once the data ingest has been performed, reports in .csv format can be generated. A large volume of schools may mean that the reporting process takes some time. When you are ready to generate reports:

1. The data ingest step must have previously been completed. (See section 3.2 Step 3 if required). Navigate to the installation folder in a console or terminal and run **naprrql.exe -report** on Windows, **naprrql -report** on Linux and Macintosh.
2. This launches the various components and services of NAPRRQL, creates two folders in the **Out** folder and generates the files listed below:
   The **school_reports** folder contains a sub folder for each school (named by school number). These contain csv files for:
   - SchoolDomainScores.csv
   - SchoolParticipation.csv
   - SchoolScoreSummaries.csv

   The system_reports folder contains the following system files:
   - SystemCodeframe.csv
   - SystemDomainScores.csv
   - SystemObjectsCount.csv
   - SystemParticipation.csv
   - SystemSchools.csv
   - SystemScoreSummaries.csv

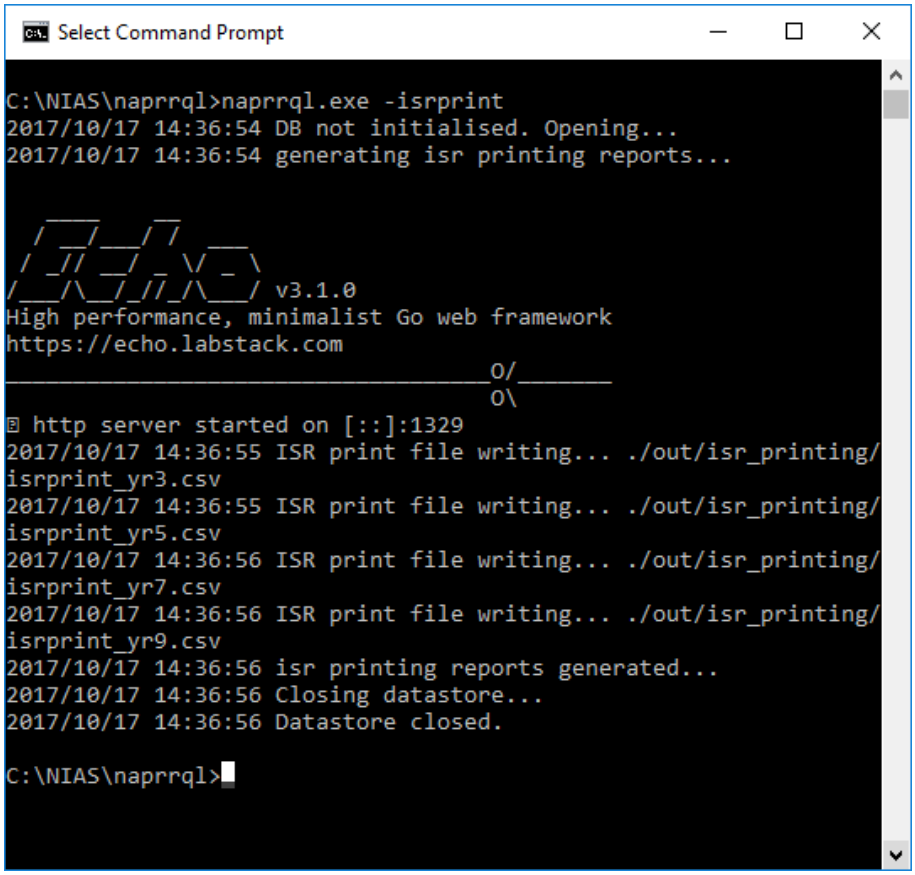


Note that if the Out folder is being viewed in File Explorer at the time of running the reports, they will not be created. Close File Explorer and try again.

3.5 ISR Printing

Once the data ingest has been performed, ISR Printing reports can be generated for each year level.

1. The data ingest step must have previously been completed. (See section 3.2 Step 3 if required). Navigate to the installation folder in a console or terminal and run **naprrql.exe -isrprint** on Windows, **naprrql -isrprint** on Linux and Macintosh. This creates csv files for each year level in a folder called **isr_printing** in the **Out** folder:
   - Isrprint_yr3.csv
   - Isrprint_yr5.csv
   - Isrprint_yr7.csv
   - Isrprint_yr9.csv



Note that either the Results Reporting or the ISR reporting csv files can be created in any sequence, as long as the ingest step has been completed first.

Section 4 - Audit facility

The NAPLAN Results Reporting data tool has a facility by which a NAPLAN Registration data file can be compared with an NAPLAN Online Results file to check for records which may be unique to each file. It compares a CSV file of Student Registration records (located in the **Napcomp/in/registration** folder) with the student records in the NAPLAN Online results XML file contained in the **Napcomp/in/results** folder, and detects which students appear only in one or the other file.

The comparison runs in two passes.

- First, records in the two files which have the same Platform Identifier (PSI) are eliminated. (For the comparison to run efficiently, users should endeavour to download from the Student Registration Management system a CSV file containing all student records, and includes their allocated PSIs.)
- Second, all remaining records from the two files are compared according to fields they have in common. The fields are specified in the naprr.toml file, under the key MatchAttributes, which contains a list of field names from the xml.RegistrationRecord struct.
  So MatchAttributes = ["FamilyName", "GivenName"], for instance, will compare the remaining records according to their family name and given name.

4.1 Using the audit facility and generating Mismatches report

1. To run the NAPRRQL audit tool, copy the required registration data file (xxx.csv) into the folder: **napcomp\in\registration**
2. The Results (xxx.XML) file should be in the **napcomp\in\results** folder.
3. Navigate to the **napcomp** folder in a console or terminal and run **napcomp.exe**. This launches the various components and services of NAPRRQL and begins the comparison of the csv and xml files.
4. Once the comparison is complete, the mismatches are output to csv files in the **napcomp\out** folder.
   - The file RegisteredButNotInResults.csv contains a listing of the students found to be unique to the NAPLAN Registration file (PSI, user-defined key, and RefId) but not in the Results and Reporting file
   - The file ResultsButNotInRegister.csv contains a listing of all the student records unique to the Results & Reporting file, but not in the Student Registration Results.

Section 5 – Querying Results and Reporting Data using GraphQL

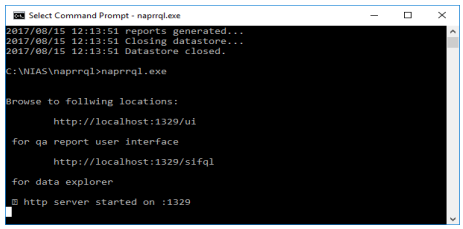
5.1 SIFQL (GraphQL) Data Explorer

NAPRRQL includes an instance of the GraphQL data explorer interface. This allows you to make queries against the underlying data-store in accordance with the SIF Schema for NAPLAN results reporting.

The full set of queries and the data elements they return is set out in the Documentation Explorer area of the data explorer user interface, a key feature is that whilst queries conform to the schema you are free to request only the fields you are interested in rather than having to receive the whole data objects.

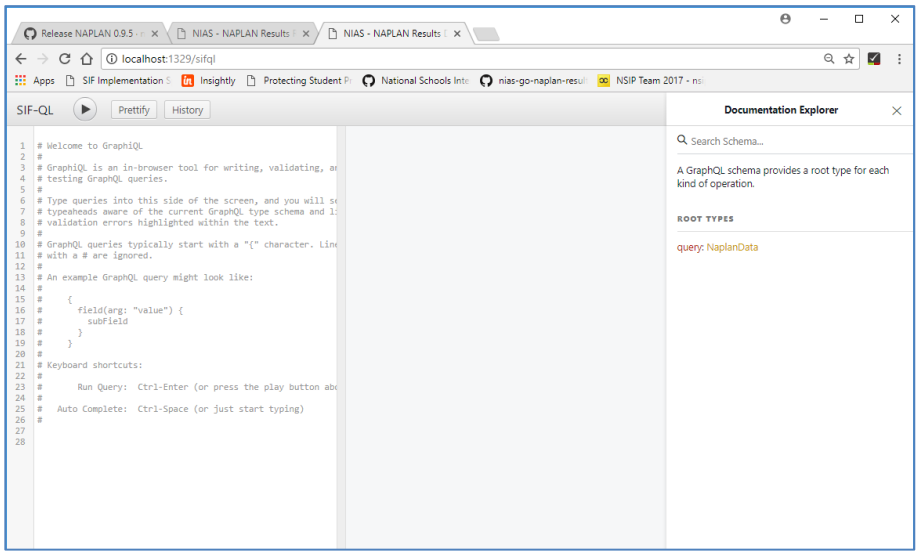
5.2 Conducting queries

1. To run the NAPRRQL tool, you must have previously run the data ingest with your Results and Reporting Dataset. (see Section 3.1 step 2)
2. Navigate to the command prompt
3. Navigate to the **naprrql** folder (e.g. C:\NIAS\naprrql)
4. Run **naprrql.exe** on Windows, **naprrql** on Linux and Macintosh



Important: The window can be minimised but should not be closed whilst running NAPRRQL. Closing the window terminates that component of the NAPRRQL reporting tool.

5. Open your browser (Google Chrome or Mozilla Firefox) and navigate to: **http://localhost:1329/sifql**
6. The following will display in the browser:



7. On your first visit to the data-explorer interface it contains a set of comments which can be safely deleted.

   Constructing queries within the interface is straightforward, but here are some samples to introduce you to the mechanism.

   These sample queries assume **naprrql** is running on its default host:port combination (localhost:1329), but if you click on the queries when **naprrql** is running you should see the queries in the explorer interface, just click on the Play button in the toolbar to run them.

   Once step 5 is complete, to open the sample queries Ctrl-click on the links below or copy and paste into a browser.

5.3 Sample NAPLAN Data Queries

NAP Tests

http://localhost:1329/sifql?query=query%20NAPTests%20%7B%0A%09tests%20%7B%0A%09%20%20TestID%0A%20%20%20%20TestContent%20%7B%0A%20%20%20%20%20%20LocalId%0A%20%20%20%20%20%20TestName%0A%20%20%20%20%20%20TestLevel%0A%20%20%20%20%20%20TestDomain%0A%20%20%20%20%20%20TestYear%0A%20%20%20%20%20%20StagesCount%0A%20%20%20%20%20%20TestType%0A%20%20%20%20%7D%0A%09%7D%0A%7D%0A&operationName=NAPTests

NAP Testlets

http://localhost:1329/sifql?query=query%20NAPTestlets%20%7B%0A%20%20testlets%20%7B%0A%20%20%20%20TestletContent%20%7B%0A%20%20%20%20%20%20LocalId%0A%20%20%20%20%20%20NAPTestLocalId%0A%20%20%20%20%20%20TestletName%0A%20%20%20%20%20%20Node%0A%20%20%20%20%20%20LocationInStage%0A%20%20%20%20%20%20TestletMaximumScore%0A%20%20%20%20%7D%0A%20%20%20%20TestItemList%20%7B%0A%20%20%20%20%20%20TestItem%20%7B%0A%20%20%20%20%20%20%20%20TestItemLocalId%0A%20%20%20%20%20%20%20%20SequenceNumber%0A%20%20%20%20%20%20%7D%0A%20%20%20%20%7D%0A%20%20%7D%0A%7D%0A&operationName=NAPTestlets

NAP TestItems

http://localhost:1329/sifql?query=query%20NAPTestItems%20%7B%0A%20%20testitems%20%7B%0A%20%20%20%20TestItemContent%20%7B%0A%20%20%20%20%20%20NAPTestItemLocalId%0A%20%20%20%20%20%20ItemName%0A%20%20%20%20%20%20ItemType%0A%20%20%20%20%20%20Subdomain%0A%20%20%20%20%20%20WritingGenre%0A%20%20%20%20%20%20ItemDescriptor%0A%20%20%20%20%20%20ReleasedStatus%0A%20%20%20%20%20%20MarkingType%0A%20%20%20%20%20%20MultipleChoiceOptionCount%0A%20%20%20%20%20%20CorrectAnswer%0A%20%20%20%20%20%20MaximumScore%0A%20%20%20%20%20%20ItemDifficulty%0A%20%20%20%20%20%20ItemDifficultyLogit5%0A%20%20%20%20%20%20ItemDifficultyLogit62%0A%20%20%20%20%20%20ItemDifficultyLogit5SE%0A%20%20%20%20%20%20ItemDifficultyLogit62SE%0A%20%20%20%20%20%2

0ItemProficiencyBand%0A%20%20%20%20%20%20ItemProficiencyLevel%0A%20%20%20%20%20%20ExemplarURL%0A%20%20%20%20%7D%0A%20%20%7D%0A%7D%0A&operationName=NAPTestItems

Score Summaries

http://localhost:1329/sifql?query=query%20NAPScoreSummaries%20%7B%0A%20%20score_summaries%20%7B%0A%20%20%20%20SummaryID%0A%20%20%20%20SchoolInfoRefId%0A%20%20%20%20SchoolACARAId%0A%20%20%20%20NAPTestRefId%0A%20%20%20%20NAPTestLocalId%0A%20%20%20%20DomainNationalAverage%0A%20%20%20%20DomainSchoolAverage%0A%20%20%20%20DomainJurisdictionAverage%0A%20%20%20%20DomainTopNational60Percent%0A%20%20%20%20DomainBottomNational60Percent%0A%20%20%7D%0A%7D%0A&operationName=NAPScoreSummaries

Students

http://localhost:1329/sifql?query=query%20NAPStudents%20%7B%0A%20%20students%20%7B%0A%20%20%20%20RefId%0A%20%20%20%20LocalId%0A%20%20%20%20StateProvinceId%0A%20%20%20%20FamilyName%0A%20%20%20%20GivenName%0A%20%20%20%20MiddleName%0A%20%20%20%20PreferredName%0A%20%20%20%20IndigenousStatus%0A%20%20%20%20Sex%0A%20%20%20%20BirthDate%0A%20%20%20%20CountryOfBirth%0A%20%20%20%20StudentLOTE%0A%20%20%20%20VisaCode%0A%20%20%20%20LBOTE%0A%20%20%20%20AddressLine1%0A%20%20%20%20AddressLine2%0A%20%20%20%20Locality%0A%20%20%20%20StateTerritory%0A%20%20%20%20Postcode%0A%20%20%20%20SchoolLocalId%0A%20%20%20%20YearLevel%0A%20%20%20%20FTE%0A%20%20%20%20Parent1LOTE%0A%20%20%20%20Parent2LOTE%0A%20%20%20%20Parent1Occupation%0A%20%20%20%20Parent2Occupation%0A%20%20%20%20Parent1SchoolEducation%0A%20%20%20%20Parent2SchoolEducation%0A%20%20%20%20Parent1NonSchoolEducation%0A%20%20%20%20Parent2NonSchoolEducation%0A%20%20%20%20LocalCampusId%0A%20%20%20%20ASLSchoolId%0A%20%20%20%20TestLevel%0A%20%20%20%20Homegroup%0A%20%20%20%20ClassGroup%0A%20%20%20%20MainSchoolFlag%0A%20%20%20%20FFPOS%0A%20%20%20%20ReportingSchoolId%0A%20%20%20%20OtherSchoolId%0A%20%20%20%20EducationSupport%0A%20%20%20%20HomeSchooledStudent%0A%20%20%20%20Sensitive%0A%20%20%20%20OfflineDelivery%0A%20%20%7D%0A%7D%0A&operationName=NAPStudents

School by School Queries

These queries return rich datasets for a single school, or for multiple schools. These queries make use of the Variables area of the UI. Select school(s) by passing the ACAR Id (ASL Id) of the school(s) required for the report. The variable AcaraIDs is an array that can contain one or more AcaraIds to identify the selected schools.

Domain Scores

http://localhost:1329/sifql?query=query%20schoolDomianScores(%24acaraIDs%3A%20%5BString%5D)%20%7B%0A%20%20domain_scores_report_by_school(acaraIDs%3A%20%24acaraIDs)%20%7B%0A%20%20%20%20Test%20%7B%0A%20%20%20%20%20%20TestContent%20%7B%0A%20%20%20

%20%20%20%20%20TestName%0A%20%20%20%20%20%20%20%20TestYear%0A%20%20%20%20%20%20%20%20TestLevel%0A%20%20%20%20%20%20%20%20TestDomain%0A%20%20%20%20%20%20%20%20StagesCount%0A%20%20%20%20%20%20%7D%0A%20%20%20%20%7D%0A%20%20%20%20Response%20%7B%0A%20%20%20%20%20%20ReportExclusionFlag%0A%20%20%20%20%20%20CalibrationSampleFlag%0A%20%20%20%20%20%20EquatingSampleFlag%0A%20%20%20%20%20%20PathTakenForDomain%0A%20%20%20%20%20%20ParallelTest%0A%20%20%20%20%20%20PSI%0A%20%20%20%20%20%20DomainScore%20%7B%0A%20%20%20%20%20%20%20%20RawScore%0A%20%20%20%20%20%20%20%20ScaledScoreValue%0A%20%20%20%20%20%20%20%20ScaledScoreLogitValue%0A%20%20%20%20%20%20%20%20ScaledScoreStandardError%0A%20%20%20%20%20%20%20%20ScaledScoreLogitStandardError%0A%20%20%20%20%20%20%20%20StudentDomainBand%0A%20%20%20%20%20%20%20%20StudentProficiency%0A%20%20%20%20%20%20%7D%0A%20%20%20%20%7D%0A%20%20%7D%0A%7D%0A&operationName=schoolDomianScores

Participation Report (queries two schools)

http://localhost:1329/sifql?query=query%20schoolParticipation(%24acaraIDs%3A%20%5BString%5D)%20%7B%0A%20%20participation_report_by_school(acaraIDs%3A%20%24acaraIDs)%20%7B%0A%20%20%20%20School%20%7B%0A%20%20%20%20%20%20LocalId%0A%20%20%20%20%20%20ACARAId%0A%20%20%20%20%20%20SchoolName%0A%20%20%20%20%20%20SchoolDistrict%0A%20%20%20%20%7D%0A%20%20%20%20Student%20%7B%0A%20%20%20%20%20%20YearLevel%0A%20%20%20%20%20%20GivenName%0A%20%20%20%20%20%20FamilyName%0A%20%20%20%20%7D%0A%20%20%20%20Summary%20%7B%0A%20%20%20%20%20%20Domain%0A%20%20%20%20%20%20ParticipationCode%0A%20%20%20%20%7D%0A%20%20%7D%0A%7D%0A&variables=%7B%0A%20%20%22acaraIDs%22%3A%20%5B%0A%20%20%20%20%2249360%22%2C%20%2249453%22%0A%20%20%5D%0A%7D&operationName=schoolParticipation

5.4 Saving queries

Note that if you find a query helpful and want to use it to produce csv reports simply save the query into a text file with a '.**gql**' extension in one of the template directories; **/school_templates** for queries that search by school, and **/system_templates** for generic data queries.

Once templates are saved in these folders re-running **naprrql** with the **-report** flag will use those queries to generate report .csv files based on the queries in the **/out** folder.

Section 6 – Support

6.1 Future functionality

Pearson/Fuji import – Coming soon

6.2 Support

Please contact NSIP directly at info@nsip.edu.au or Phone: +61 3 9910 9827 for support

To be notified of updates to the NIAS tools, subscribe to notifications on github:
https://github.com/nsip/nias2/releases