NIAS

NSIP Integration As a Service

NAPLAN Student Registration Data Validation

NAPLAN Results Reporting Data - NAPRR

Installation and User Guide v11

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
V05	27/02/2018	Minor updates
V06	17/07/2018	Added Results Reporting Validation/QA reports
V07	07/08/2018	Added updates to NAPVAL validation
V08	29/08/2018	Added Results Reporting Validation reports
V09	11/09/2018	Added Results Reporting Validation reports
V10	04/12/2019	Added to System reports, updates to saving queries
V11	26/02/2019	Added to System reports

Contents

NIAS Tools – Installation and User Guide

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.

A number of other QA and validation tools can be run using NAPRRQL. The use of these is described in detail further in this document. The table below provides a summary of what is available.

NIAS Validation Tool	Purpose	Refer to
Naprrql.exe – ingest	Ingests results and reporting data. Overwrites existing data. A prerequisite to all other functions.	Section 3
Naprrql.exe –qa	Generates csv files for validation and checking. Recommend this is run before other tools for checking prior to generating reports.	Section 6
Naprrql.exe – report	Generates reports in csv format and is prerequisite to running User Interface to view NAPLAN Results Reporting Data	Section 3
Naprrql.exe –itemprint	Generates csv file reporting item results for each student against items.	Section 6.6
Naprrql.exe –xml	Re-extracts redacted xml from Results and reporting dataset	Section 6.7
Naprrql.exe - pnpadd	Adds DAC codes	Section 6.8
Naprrql.exe –version	Displays version of NIAS in use. Used in providing support for the tool if required.	N/A

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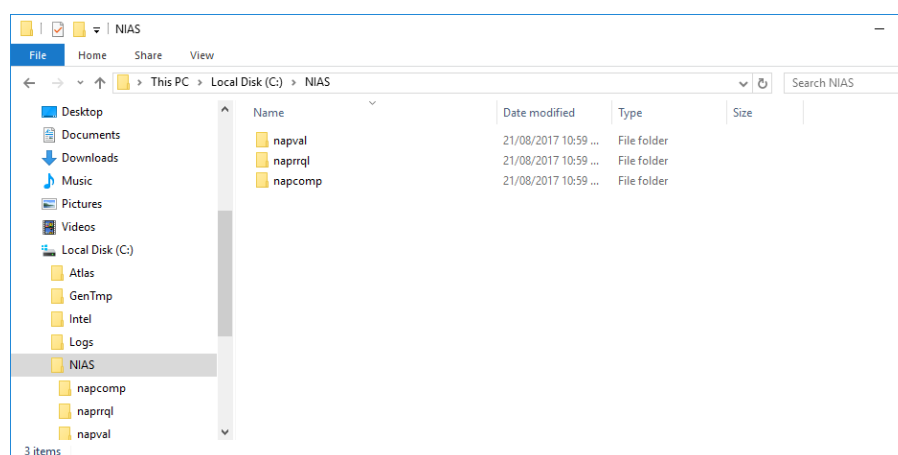
Note: Don't nest the folder structure too deeply in Windows; this can have an impact on permissible file name length.

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.

NIAS Tools – Installation and User Guide

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) <u>while it is stored on the Assessment platform</u>;
- 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.

NIAS Tools – Installation and User Guide

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 2.0. The file can be .csv or .xml format. Data files in .csv format can be converted to .xml using this tool.

The main changes to the validation process from earlier versions of NIAS (in accordance with updates to the Student Registration Data Specifications) are:

- Address Fields are no longer required for the platform, so NIAS will report an error if any address fields are populated.
    - The validation will report an error for fields that are defined in SIF but are out of scope of the Registration data set, which now includes addresses.
    - NIAS is intended for XML conversion as an output for the platform, so extra fields not defined in the CSV will not get converted to XML and therefore not uploaded to the platform.

- Visa codes have been updated

Please refer to v2.0 of the specifications for more detail.

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.

NIAS Tools – Installation and User Guide
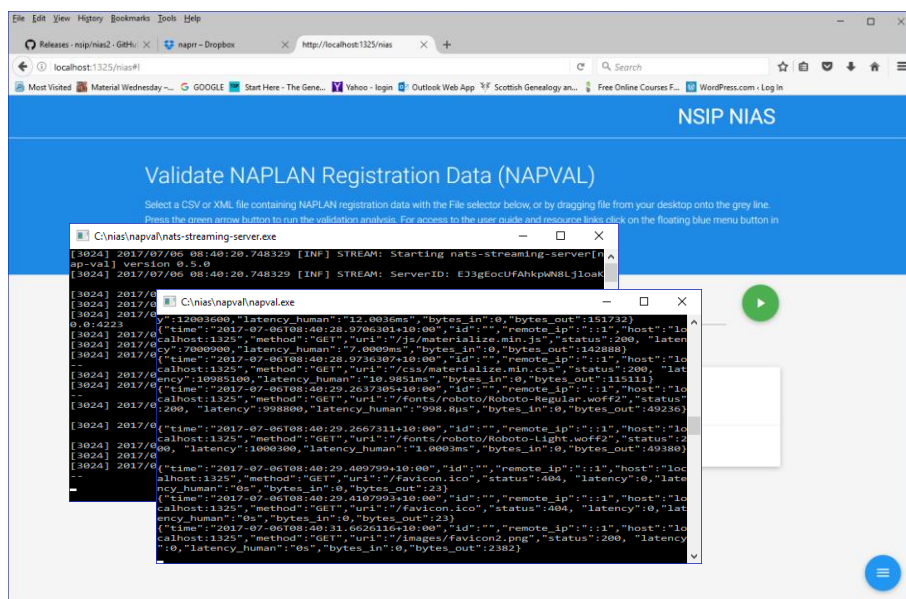
Important: These windows can be minimised but should not be closed whilst running NAPVAL.
Closing these windows terminates that component of the NAPVAL validation tool.

NIAS Tools – Installation and User Guide

3.   The NAPVAL Web UI is displayed. Users are asked to confirm privacy policy on first run – click
   ACCEPT to continue.
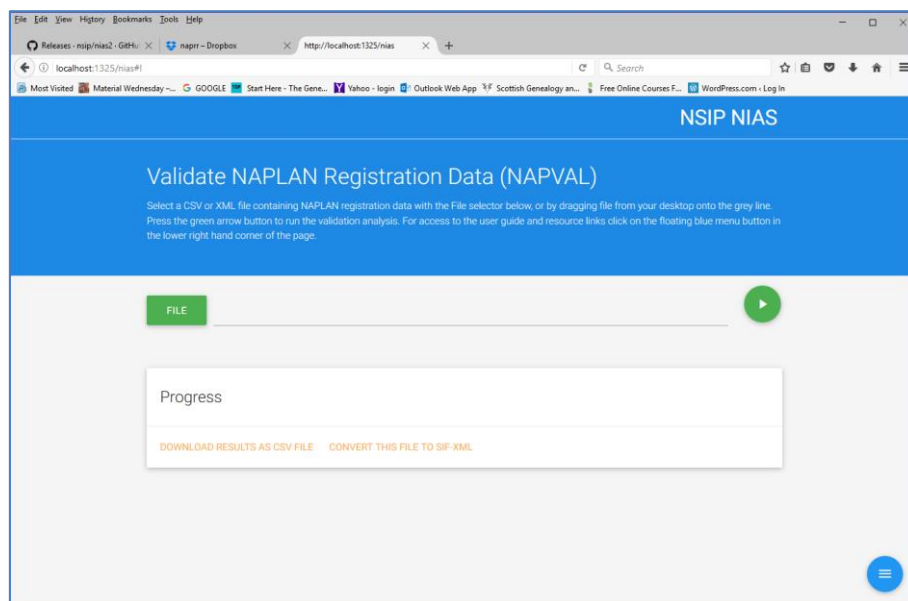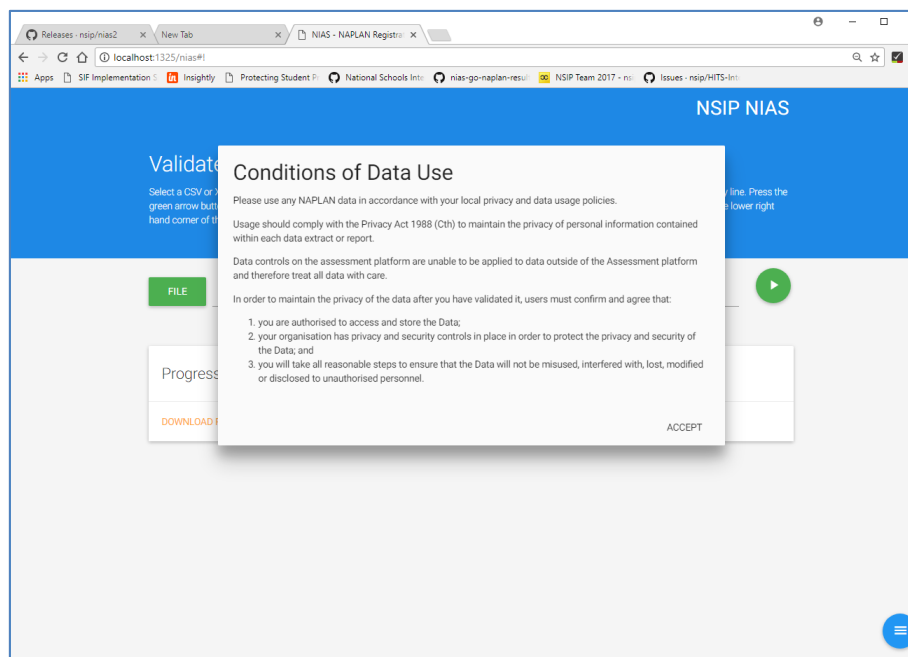
4.   Select File and choose a file to be validated from the selection box and click Open, or drop the
     file in the input location. A sample file, students.csv, is included with the software distribution in
     the NAPVAL folder. NAPVAL validates files that are either CSV or XML file formats.

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.

NIAS Tools – Installation and User Guide
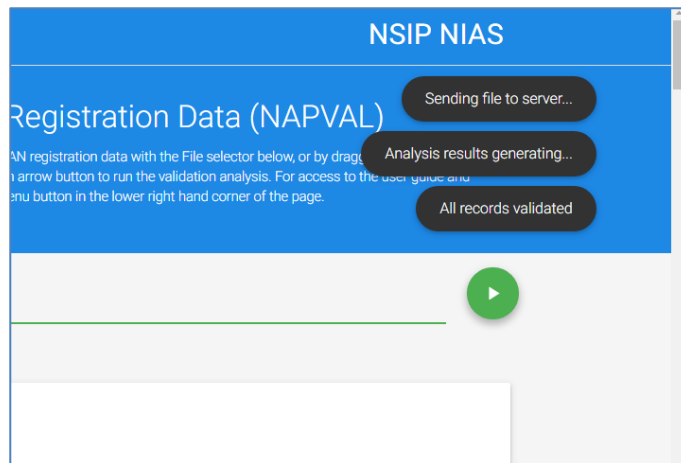
7. NAPVAL performs an initial validation, then lists a summary of the first 100 validation issues on screen.

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)
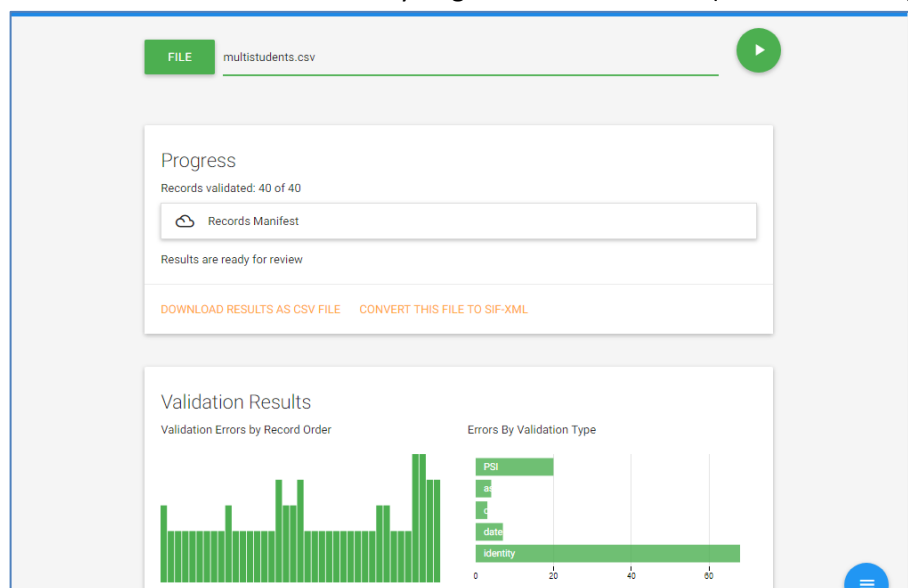

9. There is also a Records Manifest which when expanded will display the breakdown of students records per school. You can use the Manifest to confirm that the uploaded file contains the expected data.

NIAS Tools – Installation and User Guide
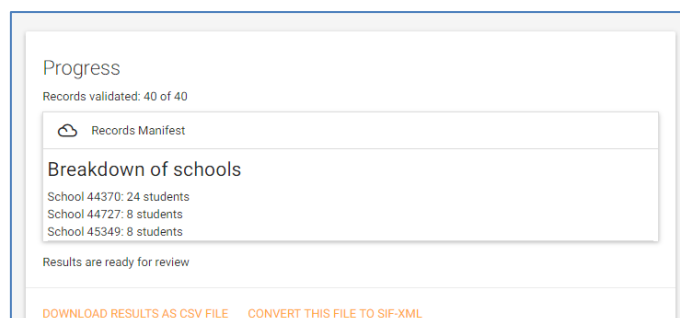
10. To view more detail on the validation errors, use one of the following controls:
   - Validation errors by record order (graph on left) or
   - Errors by validation type (graph on right)
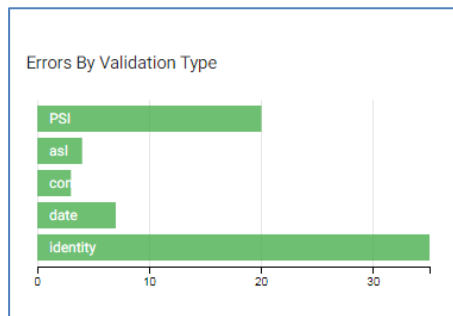
10.1 **Validation errors by record order**
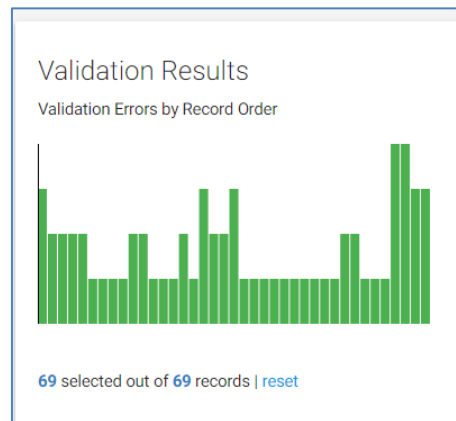
- This control allows a user the ability to narrow down the validation errors reported based on the record order.
- Move the mouse cursor over the column graph – a + symbol will appear. Drag this to select a section of the file (selected areas are green). Selecting a section will alter the errors by 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.
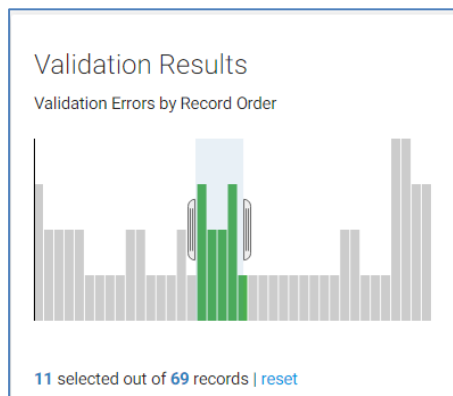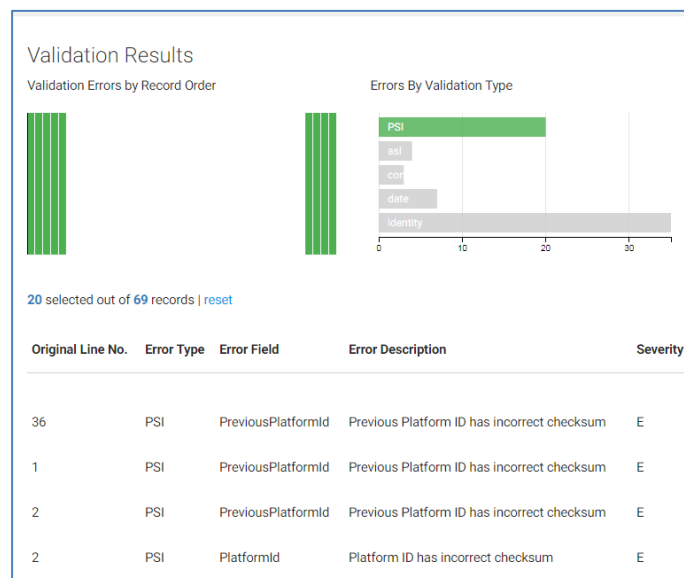

- To remove any filter on the errors reported, click on **reset.**

NIAS Tools – Installation and User Guide

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.

NIAS Tools – Installation and User Guide

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

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.

NIAS Tools – Installation and User Guide

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"]

# Legal characters for names. NOTE: this is the input to a regular
expression; if hyphens are permitted, leave - as the final character
LegalNameChars = "A-Za-z '-"

# 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.
LegalNameChars	Sets the legal characters for names. NOTE: this is the input to a regular expression; if hyphens are permitted, leave - as the final character LegalNameChars = "A-Za-z '-"

NIAS Tools – Installation and User Guide

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. For NIAS version 1.03, the ASL values were updated July 2018. (Note: ASL values can be updated by modifying values contained within the asl_schools.csv file contained within the napval\schoolslist folder)
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

NIAS Tools – Installation and User Guide

	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.
namevalid	Validates student names to confirm that they do not contain extraneous characters, which may be rejected by the Assessment Platform. The legal characters for student names are set in the LegalNameChars configuration parameter, described above.
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.

NIAS Tools – Installation and User Guide

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.

NIAS Tools – Installation and User Guide

NIAS Tools – Installation and User Guide
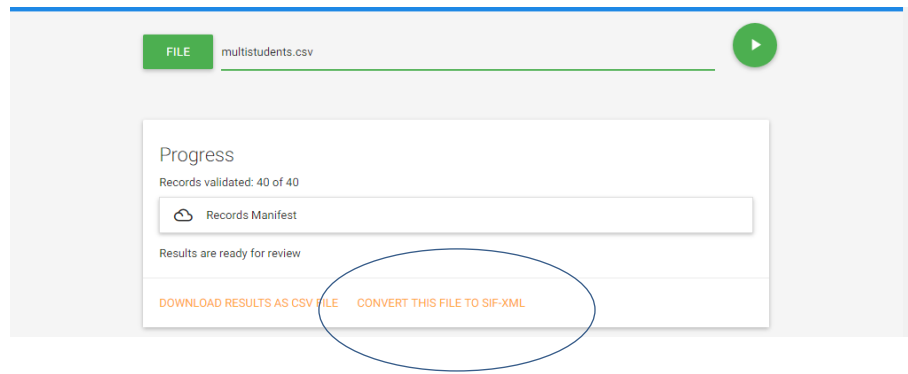
Section 3 - Using NAPRRQL

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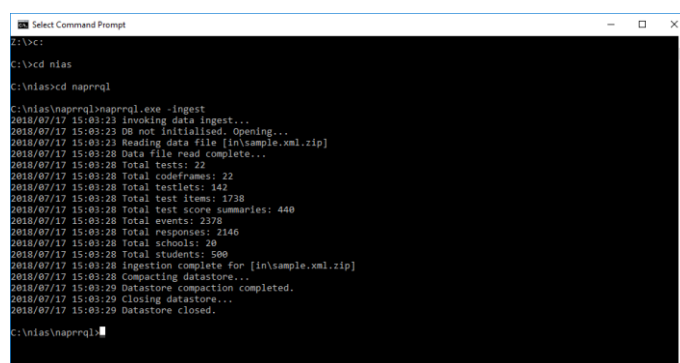
When the application is run, data is streamed from the data file in the **\in** folder and processed onto the reporting sub-system. Report generators are then run to produce .csv extracts of the data in the **\out** folder. This contains aggregate reports at the top level for all report types (score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

4. Once the ingest is complete, to generate NAPLAN Results Reports in .csv format, run **naprrql.exe -report** on Windows, **naprrql --report** on Linux and Macintosh. This step will generate an **out** folder, which will contain sub folders for school_reports and system-reports. The school_reports folder will be populated with a folder for each school and will contain individual school reports. The reports generated in the school_reports and system_reports folders are detailed in sections 6.4 and 6.5 of this document.

NIAS Tools – Installation and User Guide

5. These reports are available as csv files by viewing them in the **out** folder. Alternatively, they can be viewed using a web interface. Once the report generation is complete, to run the web interface, (still at the command prompt) 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.

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

7. The NAPLAN Results Reporting Data Web User Interface will display. Click ACCEPT to agree to the Conditions of Data Use:
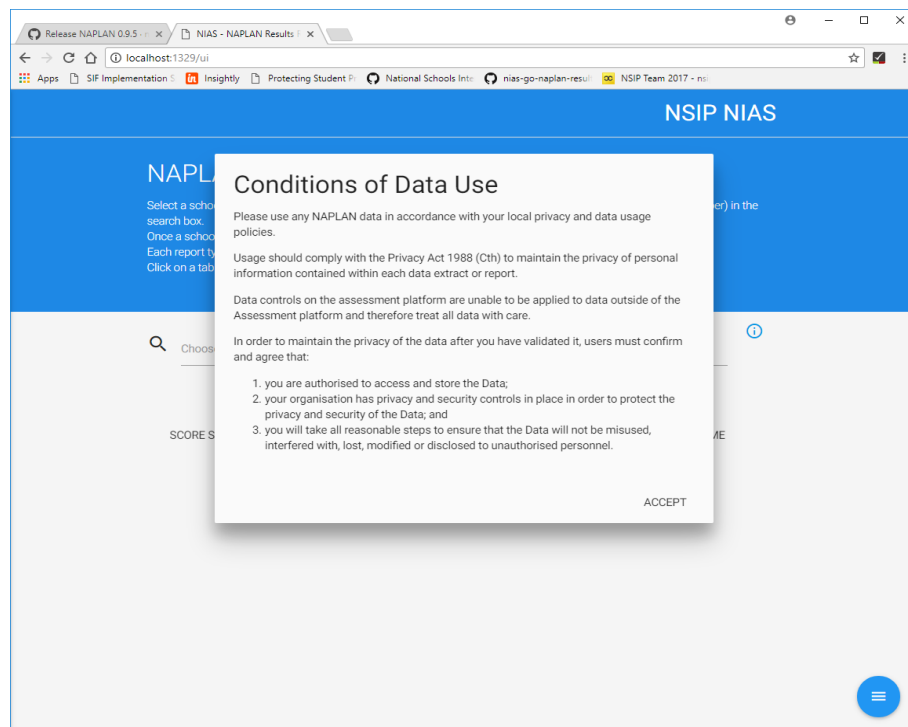
8. To begin viewing data, select a school from the search box by typing letters from the school name or the school ACARA ID and choosing the required school.

NIAS Tools – Installation and User Guide
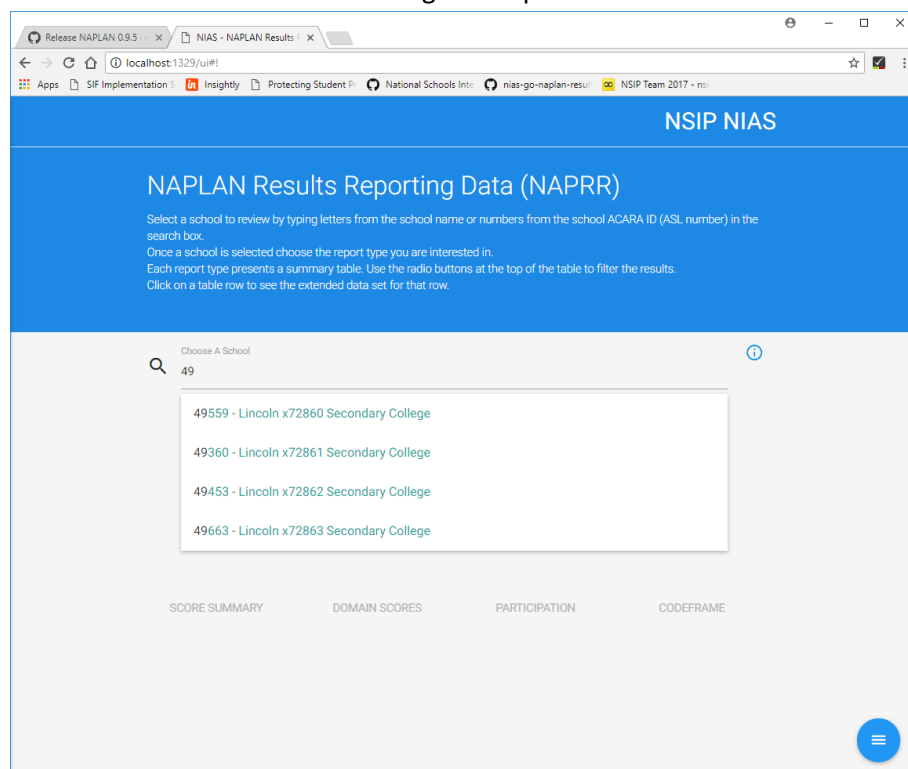
9. Once a school is selected its School Score Summary will be displayed. As the data is generated, the Domain Scores, Participation and Codeframe links will activate.

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

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

12. 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

NIAS Tools – Installation and User Guide
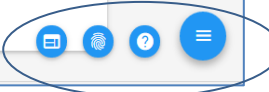
| 3 | Spelling | 401 | 405 | 403 | 419 | 383 |
| 3 | Writing | 394 | 408 | 399 | 428 | 371 |

DOWNLOAD REPORT AS CSV FILE

13. To view information about the school displayed click on the [ⓘ] button to the right of the school name

Choose A School

🔍 49360 - Lincoln x72861 Secondary College   ⓘ

SCORE SUMMARY   DOMAIN SCORES   PARTICIPATION   CODEFRAME

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

Lincoln x72861 Secondary College

23 Nicholson Street, Carnegie, VIC, 3004.

Principal: Mr Frank Mason Jr..

Contact: Mr James Mark Miller Jr.
Phone: 03 9637-2000 Ext.: 72345.

ACARA (ASL) Id: 49360
Local Id: x72861
State/Province Id: x72861
District: Southern Metropolitan Region
Campus: Y
Independent School: Y
School Sector: NG
School Type: Pri/Sec
Year Levels: 6 7 8 9 10 11 12
Student Count:
.

ARIA: 1.0
NG Systemic Status: S
Operational Status: O
Co-Ed Status: C
LGA: Cardinia
Federal Electorate: 216
Religious Affiliation: 2171
Geographic Location: 1

Grid Location: 23.9876, -98.8765

NSIP NIA

N Results Reporting Data (NAPRR)

o review by typing letters from the school name or numbers from the school ACARA ID (ASL number) in the search
selected choose the report type you are interested in.
presents a summary table. Use the radio buttons at the top of the table to filter the results.
ow to see the extended data set for that row.

chool   ⓘ
ncoln x72861 Secondary College

IMARY   DOMAIN SCORES   PARTICIPATION   CODEFRAME

Score Summary

○ Yr 3   ○ Yr 5   ○ Yr 7   ○ Yr 9

| in | School Average | Jurisd. Average | National Average | Top National 60% | Bottom National 60% |
| mar and uation | 401 | 407 | 401 | 412 | 380 |

NIAS Tools – Installation and User Guide

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 <u>removed</u> 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 reporting. Note that the ingest process only needs to be done once unless the data changes.
5. 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.
6. Once the ingest is complete, to generate NAPLAN Results Reports in .csv format, run **naprrql.exe -report** on Windows, **naprrql --report** on Linux and Macintosh.
7. 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. Alternatively, the csv files can be viewed in the **out** folder.
8. **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.**
9. Open your browser (Google Chrome or Mozilla Firefox) and navigate to: **http://localhost:1329/ui**
10. The NAPLAN Results Reporting Data Web User Interface will display.
11. 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.
12. 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.
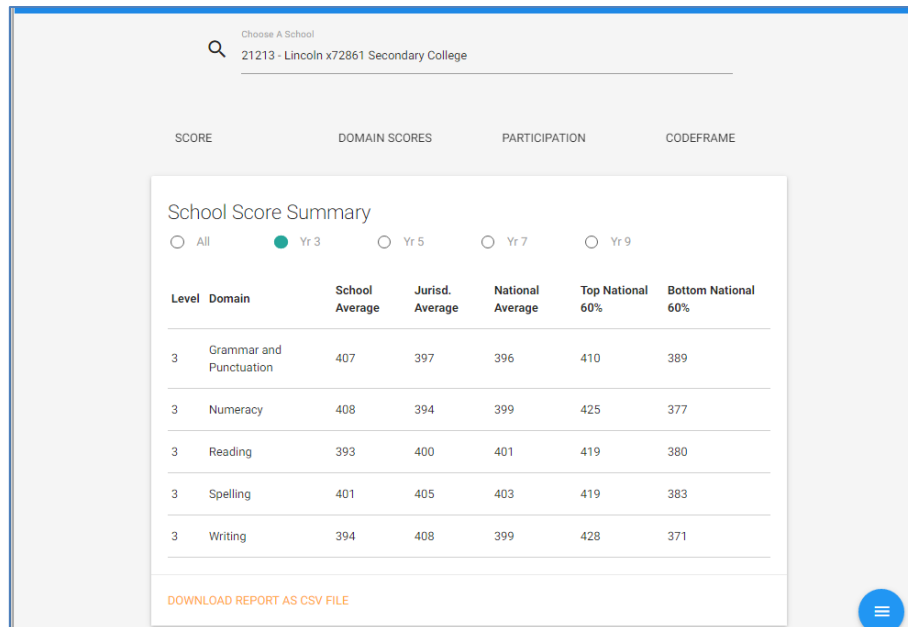
The displayed report can be exported by clicking on the link at the bottom of the report

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.

NIAS Tools – Installation and User Guide

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.

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

NIAS Tools – Installation and User Guide

Click CLOSE to return to the previous screen.

NIAS Tools – Installation and User Guide

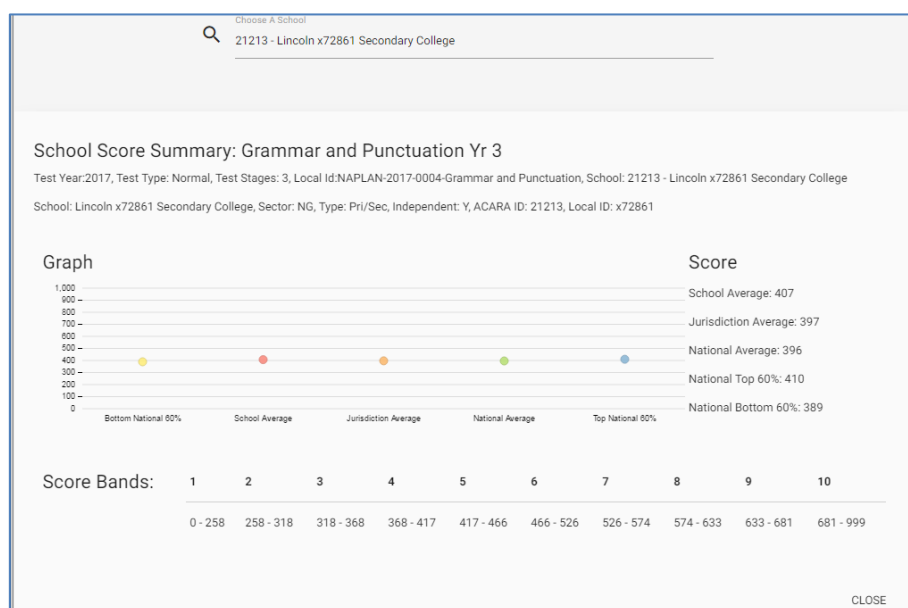## Student Participation Report

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.

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.

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.

NIAS Tools – Installation and User Guide

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.

NIAS Tools – Installation and User Guide

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:

NIAS Tools – Installation and User Guide
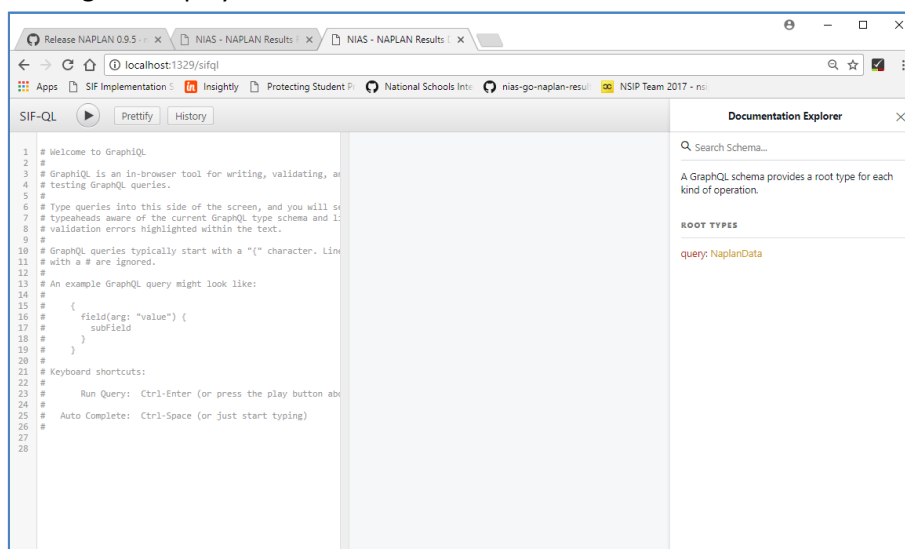
7. On your first visit to the data-explorer interface it contains a set of comments which can be safely deleted.

   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

NIAS Tools – Installation and User Guide

0ItemProficiencyBand%0A%20%20%20%20%20%20ItemProficiencyLevel%0A%20%20%20%2
0%20%20ExemplarURL%0A%20%20%20%20%7D%0A%20%20%7D%0A%7D%0A&operation
Name=NAPTestItems

Score Summaries

http://localhost:1329/sifql?query=query%20NAPScoreSummaries%20%7B%0A%20%20score_summ
aries%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=NAPScoreSum
maries

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%20StateProvi
nceId%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%20CountryOfBirt
h%0A%20%20%20%20StudentLOTE%0A%20%20%20%20VisaCode%0A%20%20%20%20LBO
TE%0A%20%20%20%20AddressLine1%0A%20%20%20%20AddressLine2%0A%20%20%20%2
0Locality%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%20Homegro
up%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%2
0%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&operati
onName=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%5
D)%20%7B%0A%20%20domain_scores_report_by_school(acaraIDs%3A%20%24acaraIDs)%20%7B%0
A%20%20%20%20Test%20%7B%0A%20%20%20%20%20%20TestContent%20%7B%0A%20%20%20

NIAS Tools – Installation and User Guide

[%20%20%20%20%20TestName%0A%20%20%20%20%20%20%20TestYear%0A%20%20%20%20%20%20%20%20TestLevel%0A%20%20%20%20%20%20%20TestDomain%0A%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](http://localhost)

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](http://localhost)

5.4   Saving queries and generating reports

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; **/reporting_templates/schools** for queries that search by school, and **/reporting_templates/system** 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.

Each query has a corresponding map.csv file. Normally this defines the header fields for the CSV output of a query, and the corresponding fields to be retrieved from the query, in dot notation; e.g. Student.LocalId means "the Local ID field of the student record".

The system can also generate fixed width reports, as with nswWritingPearsonY3. In that case, the first line of the file is "FixedFormat", the third line gives the fields to be retrieved. The second line is the width in characters for each field to be output, rather than its headers; if the character count is preceded by a zero, the field is zero-padded. Thus, 3,4,09 means to allocate three characters for the first field, four for the second, and nine for the third, with zero padding.

NIAS Tools – Installation and User Guide

Section 6 – NAPLAN Results Reporting Validation and QA Reports

NIAS produces a number of reports for validation of the results and reporting dataset. The tables in section 6.2 describe these reports.

6.1 Generating Results Validation reports

1. The data ingest step must have previously been completed. (See section 3.2 Step 3 if required). Navigate to the **naprrql** folder (e.g. C:\NIAS\naprrql)
2. Run **naprrql.exe -qa** on Windows, **naprrql –qa** on Linux and Macintosh.
3. This launches the various components and services of NAPRRQL, creates a **qa** folder in the **out** folder and generates the files listed below:

6.2 QA Reports

itemWritingPrinting.csv	
Purpose	• Reports one row of response data per student per test item presented within a NAPLAN test for the Writing domain only • Detailed information which can be used for ingest into data analysis systems
What to expect	• Very large report • Includes a column for each subscore
Look out for	

orphanEvents.csv	
Purpose	• Reports any NAPStudentEvents whose corresponding school does not have a SchoolInfo object in the data set (no details have been provided about the school, or the school does not exist)
What to expect	• Ideally this report should be empty
Look out for	• Any entry in this report needs to be investigated

orphanScoreSummaries.csv	
Purpose	• Reports any score summaries whose corresponding school does not have a SchoolInfo object in the data set (no details have been provided about the school, or the school does not exist)
What to expect	• Ideally this report should be empty
Look out for	• Any entry in this report needs to be investigated

orphanStudents.csv	
Purpose	• Reports any students whose corresponding school does not have a SchoolInfo object in the data set (no details have been provided about the school, or the school does not exist)
What to expect	• Ideally this report should be empty
Look out for	• Any entry in this report needs to be investigated

NIAS Tools – Installation and User Guide

qaCodeframeCheck.csv	
Purpose	• Reports all tests, testlets and items referenced in responses that are not included in the Codeframe
What to expect	• Ideally this report should be empty
Look out for	• Tests / Testlets / TestItems should not be responded to if they are not part of the codeframe: any entry in this report needs to be investigated

qaSchools.csv	
Purpose	• Reports one row per school, summary of school information as well as data from student registration information and results information • Includes total registered students and summary of student count by Year level • Includes total test attempts by year level and domain • Includes counts of students by participation status • Includes count of disruptions
What to expect	• All schools for the sector should be listed • Can check response counts against expected students registered per test level
Look out for	• Discrepancies between number of students registered at each year level for each test domain; reasonable number of students exempt, absent, experiencing disruptions etc.

systemCodeframe.csv	
Purpose	• Reports all Items against Testlets and Tests in the Codeframe • Includes Item difficulty, correct answer, item type • Detailed information which can be used for ingest into data analysis systems
What to expect	• Large report • One row for every Item in Codeframe
Look out for	

systemGuidCheck.csv	
Purpose	• Reports any GUID that references the wrong kind of object, or doesn't reference any object included in the data set
What to expect	• Ideally this report should be empty
Look out for	• If any data appears, report will show the correct kind of object the GUID should point to. Any entry in this report needs to be investigated.

systemResponses.csv	
Purpose	• Reports one row for every student per school registered for a test domain, along with information on their response if available

NIAS Tools – Installation and User Guide

	• Includes participation code, path taken for domain and raw score
	• Detailed information which can be used for ingest into data analysis systems
What to expect	• Very large report
	• Will include a row for every student registration for all tests, but not all rows will contain responses
Look out for	

systemScoreSummaries.csv	
Purpose	• Reports a row for every score summary included in the data set, sorted by school, domain and year level
What to expect	• Use to check expected year levels
	• 2017/18 writing may be included more than once, as score summaries are per test rather than test domain
Look out for	• Any school without an entry for a domain–year level pair (other than Yr 3 Writing)

systemExtraneousCharactersStudents.csv	
Purpose	• Reports a row for every student whose name contains a character other than a letter, an apostrophe, a hyphen, or space
What to expect	• Ideally this report should be empty
Look out for	• Check for any students whose names may cause SSSR report to crash

systemStudentEventAcaraIdDiscrepancies.csv	
Purpose	• Reports a row for every test administered at a different school from where the student was enrolled
What to expect	• Ideally this report should be empty
Look out for	• Check for any students whose school of enrolment and school administering the test conflict. If a student has been registered against two schools, the first school of registration will be used as the control.

systemStudentTestYearLevelDiscrepancies.csv	
Purpose	• Reports a row for every test administered at a different test level from the year level that the student was enrolled in
What to expect	• Ideally this report should be empty
Look out for	• Check for any students whose year level conflicts with the test level they were administered.

systemMissingTestlets.csv	
Purpose	• Reports a row for every student response set with a participation status of P, which contain less testlets (according to the ParallelTest field of the student response set object) than

NIAS Tools – Installation and User Guide

	are expected for the given test level and test domain. (For 2018, these are: three for Reading and Spelling; three for Numeracy Yrs 3 and 5; four for Numeracy Yrs 7 and 9; one form Grammar & Punctuation and Writing.)
What to expect	• Ideally this report should be empty
Look out for	• Check for any students that have not been presented the full count of testlets expected. Check the test disruptions and total lapsed time for the test event, as potential explanations.

6.3 QA\Error_reports

Reports are generated in this folder as a result of using the **-qa** option in NIAS

itemExpectedResponses.csv	
Purpose	• Reports one row per student per test • Presently up to 3 Testlets per test • Lists the number of items per testlet that a student is expected to have answered (according to the codeframe), the number of correct/incorrect/not attempted and not presented items in that testlet; and any discrepancies between the number of items expected and referenced in the response • Accounts for any cases where all item responses in the testlet are missing. To highlight these the report injects a dummy "empty testlet report" item response into the testlet which is displayed in the report error. Any students with partial results in their PRD will be found.
What to expect	• A row per student response per test • At least one testlet per test populated
Look out for	• No items in the Not In Path category • Listings of discrepancies ("Expected not found, Found not expected") to be empty square brackets ("[]") • Number of expected items per testlet to equal the correct/incorrect/not attempt/not presented items per testlet

systemItemCounts.csv	
Purpose	• Reports the number of times an item is in a response at all
What to expect	• A row for every item in the codeframe
Look out for	• High counts for substitute items • Zero counts for any item

systemObjectFrequency.csv	
Purpose	• Reports for every student the number of events they have recorded, the number of events with participation status P/R/S (Participated/Refused/Sanctioned Abandonment)—which are expected to have produced responses; and the number of responses

NIAS Tools – Installation and User Guide

	• Also includes P/R/S events without responses, and responses without P/R/S events
What to expect	• A row for every unique student in the data set, independent of school
Look out for	• Check for discrepancies the number of events with participation status P/R/S and the number of responses • Check for any instances of a student participating in more than five distinct events (as identified by test level plus test domain)

systemParticipationCodeImpacts.csv	
Purpose	• Reports tests for which the response contents are unexpected based on the participation code
What to expect	• Ideally this report should be empty

Look out for	• "Adaptive pathway without student undertaking test" if:
	     o participation code = S, P
	     o PathTakenForDomain or ParallelTest are not empty
	• "Scored test with status other than P or R" if:
	     o participation code ≠ P, R
	     o RawScore or ScaledScore is not empty
	• "Non-zero score with status of R" if:
	     o participation code = R
	     o RawScore is not empty
	     o RawScore ≠ 0
	• "Unscored test with status of P or R"
	     o participation code = P, R
	     o RawScore or ScaledScore is empty

systemRubricSubscoreMatches.csv	
Purpose	• Reports item/response pairs in which the rubric types and subscore types do not match
What to expect	• Ideally this report should be empty
Look out for	• Where there are SubscoresNotDefined which are subscores with no corresponding rubric, or that are not in the list of 10 expected rubric names • Where there are RubricsNotScored which are rubrics with no corresponding subscore in the item response

systemParticipationCodeItemImpacts.csv	
Purpose	• Reports items for which the item responses are unexpected based on the participation code, focusing on score at test level, testlet level and item level
What to expect	• Ideally this report should be empty

Look out for	• "Response captured without student writing test" if:
	     o participation code ≠ S, P
	     o Response is not empty
	• Scored test with status other than P or R if:
	     o participation code ≠ P, R

	o TestletScore, ItemScore or Subscores are not empty
	• Non-zero Scored test with status of R if:
	  o Participation code = R
	  o TestletScore, ItemScore or Subscores is not empty
	  o TestletScore, ItemScore or Subscores ≠ 0
	• Missing testlet score with status of P or R if
	  o Participation code = P, R
	  o TestletScore is empty
	  o This error is not as serious
	• Unscored test with status of P or R
	  o Participation code = P, R
	  o ItemScore is empty
	• Unscored writing test with status of P
	  o ParticipationCode = P
	  o TestDomain = Writing
	  o Subscore is empty

systemTestAttempts.csv	
Purpose	• Reports all sanctioned abandonments for students in schools for sector
What to expect	• Small number of rows
Look out for	

systemTestCompleteness.csv	
Purpose	• Reports for each test per year level per school in sector counts for Present and Sanctioned Abandonment, and discrepancies between attempts (events with status P or R) and responses
What to expect	• Row per school per domain per test level
Look out for	• List of attempts with no response should be empty for all rows • List of responses with no attempts should be empty for all rows • P_Attempts + S_Attempts + R_Attempts should add up to Responses

systemTestIncidents.csv	
Purpose	• Reports all non-empty test disruptions by student, school and domain
What to expect	• Small number of rows for disruptions impacting individual students • Larger number of rows for disruptions impacting cohort of students
Look out for	• Schools with high proportion of disruption

systemTestTypeImpacts.csv	
Purpose	• Reports tests for which the response contents are unexpected based on the test domain

NIAS Tools – Installation and User Guide

What to expect	• Ideally this report should be empty

Look out for	• "Writing test with adaptive structure" if
	    ○ TestDomain = Writing
	    ○ PathTakenForDomain or ParallelTest is not empty
	• "Non-Writing test with non-adaptive structure" if
	    ○ TestDomain ≠ Writing
	    ○ PathTakenForDomain or ParallelTest is empty

systemTestTypeItemImpacts.csv	
Purpose	• Reports items for which the item responses are unexpected based on the test domain, focusing on score at test level, testlet level and item level
What to expect	• Ideally this report should be empty

Look out for	• "No subscores for Writing Test" if
	    ○ TestDomain = Writing
	    ○ Subscores is empty
	• Subscores for non-writing test if
	    ○ TestDomain ≠ Writing
	    ○ Subscores is not empty

6.4 School Reports

School_reports: Reports are generated in this folder after using the **-report** option in NIAS. (Section 3).

schoolDomainScores.csv	
Purpose	• Report contains line per student in the school for each domain containing domain band, raw and scaled scores
What to expect	• One record for each test where response is available
Look out for	• Zero scores if Participation = Refused • Blank/Null if Participation = Sanctioned Abandonment

schoolParticipation.csv	
Purpose	• Reports participation status for each student for each domain within the school
What to expect	• One row per student
Look out for	

schoolScoreSummaries.csv	
Purpose	• Reports for each year level and domain averages for School, Jurisdiction and National • Reports top and bottom national 60%
What to expect	• One row per test name (made up of test level and domain name) for the school
Look out for	

NIAS Tools – Installation and User Guide

6.5 System Reports

System_reports: Reports are generated in this folder as a result of using the **-report** option in NIAS. In the following,

- reports prefixed with qld are specific to Queensland,
- reports prefixed with nsw are specific to NSW,
- reports prefixed with qcec are specific to the QCEC,
- reports prefixed with act are specific to ACT,
- reports prefixed with sa are specific to South Australia.

systemCodeframe.csv	
Purpose	• Reports for each year level and domain, Testlet information including Name, Location, Node, Maximum Score and Items, including difficulty Maximum score, difficulty and Item Type
What to expect	
Look out for	

systemDomainScores.csv	
Purpose	• Reports raw and scaled scores for each student, year level and domain
What to expect	• One row per student per test domain
Look out for	

systemObjectsCount.csv	
Purpose	• Summary report showing counts for: schools, students, test events, student responses, tests, testlets, test items, codeframes & score summaries
What to expect	
Look out for	

systemParticipation.csv	
Purpose	• Reports participation codes for each student in each school by domain
What to expect	
Look out for	

systemSchools.csv	
Purpose	• Reports school registration details, including school type, sector, principal name and websites
What to expect	
Look out for	• Student count will only be populated if it has been submitted in Registration

NIAS Tools – Installation and User Guide

systemScoreSummaries.csv	
Purpose	• Reports for each school the Local Test Name and averages for School, Jurisdiction and National
What to expect	
Look out for	

qldStudent.csv	
Purpose	• Basic registration information for each student by school in TAA, following QLD TAA format requirements
What to expect	
Look out for	

qldStudentScore.csv	
Purpose	• Reports Student ID Domain, Year level, raw and Scaled scores, Band Participation status for each student, following QLD TAA format requirements
What to expect	
Look out for	

qldTestData.csv	
Purpose	• Reports item information per year level and domain, includes sequence and item type, following QLD TAA format requirements
What to expect	
Look out for	

isrPrinting.csv	
Purpose	• Reports for student including local ID, name, scores for each domain, mean scores • Can be used to facilitate merge with paper records to support printing of hardcopy ISRs.
What to expect	• A record for each student
Look out for	

isrPrintingExpanded	
Purpose	• Reports for student including local ID, name, and, for each domain: scaled scores for the student, scaled score standard deviations for the student, mean scores (across the school), pathways for test responses, and full demographic information for each student • Can be used to facilitate merge with paper records to support printing of hardcopy ISRs.
What to expect	• A record for each student
Look out for	

NIAS Tools – Installation and User Guide

nswItemDescriptors.csv	
Purpose	• Reports the item descriptor for each item in the codeframe
What to expect	
Look out for	

nswItemPrinting.csv	
Purpose	• Reports item responses for each item responded to, against the student PSI, following NSW TAA format requirements
What to expect	• A record for each item response
Look out for	

nswPrint.csv	
Purpose	• Reports the domain scores, participation status, and DAC/PNP codes for each test sat by a student, following NSW TAA format requirements
What to expect	• A record for each student, including all test domain responses
Look out for	

nswPrintAll.csv	
Purpose	• Reports all item responses for all tests sat by a student, following NSW TAA format requirements
What to expect	• A record for each student, including all item responses in all tests
Look out for	• Very large number of columns: the report has a fixed number of columns, and allows 30 items per testlet, with 3 testlets per test (4 for Yr 7 and 9 Numeracy).

nswWritingScripts.csv	
Purpose	• Reports writing scripts for all Writing tests sat by students, following NSW TAA format requirements
What to expect	• A record for each student, including their writing script.
Look out for	• HTML and punctuation in the writing script field. • Escaped quotation marks and quotation marks around the entire field, if necessary, to comply with CSV requirements.

nswWritingPearsonY3.csv	
Purpose	• Output the writing results of year 3 students, following the fixed-width format prescribed by Pearson for processing. Because no students are registered in the platform for Year 3 Writing, the report outputs demographic information for Year 3 Numeracy students instead, to save effort.
What to expect	• A record for each student, with fixed width fields rather than comma-delimited fields.
Look out for	• Students enrolled in Writing but not Numeracy.

NIAS Tools – Installation and User Guide

	• There should be no writing results included in the report, since none are present in the RRD file: the file is to be used as a template to be populated by merging the writing results in, out of band.

nswWritingPearsonY5.csv	
Purpose	• Output the writing results of year 5 students, following the fixed-width format prescribed by Pearson for processing
What to expect	• A record for each student, with fixed width fields rather than comma-delimited fields.
Look out for	• There should be no writing results included in the report, since none are present in the RRD file: the file is to be used as a template to be populated by merging the writing results in, out of band.

nswWritingPearsonY7.csv	
Purpose	• Output the writing results of year 7 students, following the fixed-width format prescribed by Pearson for processing
What to expect	• A record for each student, with fixed width fields rather than comma-delimited fields.
Look out for	• There should be no writing results included in the report, since none are present in the RRD file: the file is to be used as a template to be populated by merging the writing results in, out of band.

nswWritingPearsonY9.csv	
Purpose	• Output the writing results of year 9 students, following the fixed-width format prescribed by Pearson for processing
What to expect	• A record for each student, with fixed width fields rather than comma-delimited fields.
Look out for	• There should be no writing results included in the report, since none are present in the RRD file: the file is to be used as a template to be populated by merging the writing results in, out of band.

qcecPrintAll.csv	
Purpose	• Reports all item responses for all tests sat by a student, following QCEC format requirements
What to expect	• A record for each student, including all item responses in all tests
Look out for	• Very large number of columns: the report has a fixed number of columns, and allows 30 items per testlet, with 3 testlets per test (4 for Yr 7 and 9 Numeracy).

NIAS Tools – Installation and User Guide

actSystemDomainScores.csv	
Purpose	• Reports raw and scaled scores for each student, year level, domain, parallel path, and participation
What to expect	• One row per student per test domain
Look out for	

systemPNPEvents.csv	
Purpose	• Reports on all students with at least one PNP code for an event.
What to expect	• One row for each test event which has at least one associated PNP code.
Look out for	

saHomeschooledTests.csv	
Purpose	• Reports on test attempt status for home schooled students
What to expect	• A record for each student, including test attempt status and Reporting school details.
Look out for	

6.6 Item Printing

To generate **itemResults.csv**, run **naprrql.exe -itemprint** on Windows, **naprrql --itemprint** on Linux and Macintosh. This generates a CSV report of all individual item responses for all domains except for Writing (which is generated instead in the QA report itemWritingPrinting.csv). There is one row for each item responded to by a student.

This creates **ItemResults.csv** within the folder **itemprinting** (under the **out** folder).

itemPrinting.csv	
Purpose	• Reports one row of response data per student per test item presented within a NAPLAN test • Detailed information which can be used for ingest into data analysis systems
What to expect	• Very large report • This report excludes writing
Look out for	

NIAS Tools – Installation and User Guide

6.7 XML

Note: This functionality generates modified RRD XML file/s and it is recommended to be used only after reading the below guidance.

This function is used to re-extract redacted xml from the Results and Reporting dataset as both a single file per ingest (typically a single RRD) AND a separate XML file per school (based on ACARAID).

Run **naprrql.exe -xml** on Windows, **naprrql --xml** on Linux and Macintosh.

naprrql —xml: Outputs all ingested NAPRR records in XML format. This outputs both a single file for the entire sector, as out/xml/sif.xml, and one file per school, as out/xml/ACARID/sif.xml . All school-specific XML files include the same codeframe information, as well as student and response data specific to the school; so the school-specific XML files are self contained.

<u>IMPORTANT:</u> If you want the XML output to reflect the full contents of the input XML file, you *must* remove all entries from the XMLFilter element in the naprrql.toml file (the supplied naprrql.toml file contains entries to demonstrate the redaction functionality).

The XML follows the SIF/XML schema just like the output of the National Assessment Platform, **but it has the following differences**:

- There are no container elements, such as StudentPersonals or NAPTests (although the container elements are not used consistently in the source XML anyway).
- No empty elements will be generated: elements like "<StateProvince />" and "<GridLocation xsi:nil="true" />" will simply not appear in the output.
- Empty list elements (which are against the SIF spec to begin with), such as "<ItemResponseList/>", will not appear in the output.

You can set elements to be redacted in the XMLFilter element in the **naprrql.toml** file. The XMLFilter element is an array of paths of elements to be redacted, in JSON Dot notation; e.g. "NAPTest.TestContent.DomainBands.Band1Upper" (corresponding to the XPath "NAPTest/TestContent/DomainBands/Band1Upper"), "NAPTestlet.TestItemList.TestItem.0.TestItemLocalId" (corresponding to the XPath "NAPTestlet/TestItemList/TestItem[1]/TestItemLocalId").

The syntax does not currently support redaction across multiple list entries; a workaround which will work in simple cases is to insert multiple entries, for the likely number of list entries applied (e.g. "NAPTestlet.TestItemList.TestItem.0.TestItemLocalId", "NAPTestlet.TestItemList.TestItem.1.TestItemLocalId", "NAPTestlet.TestItemList.TestItem.2.TestItemLocalId"…)

The elements to be redacted follow the GraphQL definition of the objects, rather than the XML definition (e.g. RegistrationRecord.FamilyName, not StudentPersonal.PersonInfo.Name.FamilyName;

NIAS Tools – Installation and User Guide

consult the GraphQL schema under gql_schemas/naplan_schema.graphql for specifics.)

We have provided some sample values in the **naprrql.toml** provided with the downloaded program. **IMPORTANT:** If you want the XML output to reflect the full contents of the input XML file, you *must* remove all entries from the XMLFilter element in the naprrql.toml file (the supplied naprrql.toml file contains entries to demonstrate the redaction functionality).

Redaction involves simply setting an element value to empty; the element will be rendered in XML as empty if it is a mandatory element, and will be omitted if it is optional. The resulting XML may not validate correctly for mandatory elements, as empty values violate XML schema types; this is already an issue for SIF as it deals with privacy redaction. **Make sure any downstream consumers of the XML are alerted about the redaction of mandatory elements.**

Redaction of XML files is usually done via XSLT. The approach taken by NIAS of redacting source JSON records has the advantage of being performant over a large set of data. XSLT 1.0 processors, by contrast, will run into trouble attempting to process files larger than 500 MB, and it will be necessary to split any larger files into smaller components before processing them in XSLT. (This approach was taken in https://github.com/nsip/nias_writing_extract_test.)

However, the approach taken here only deals with simple redaction paths, as seen; it does not deal with redacting a match across an arbitrary number of list entries, for example. XSLT processing may be added to NIAS at a later date if required.

6.8 DAC/PNP codes

Note: This functionality modifies the RRD and is only recommended to be used with guidance from the NSIP development team.

This function adds DAC codes from the CSV file "FILENAME.CSV" into the RRD extract currently ingested into NIAS, and outputs the patched RRD extract as an XML file, using the same conventions as the -xml option. (The XML file is output as out/sif.xml.) The CSV file is expected to be the report out/system/systemPNPEvents.csv generated by naprrql -report, or a subset of rows from that report.

Run **naprrql.exe –pnpadd FILENAME.CSV** on Windows, **naprrql -- pnpadd FILENAME.CSV** on Linux and Macintosh.

NIAS Tools – Installation and User Guide

6.8 Reports available via NAPRRQL user interface

These reports are available using the NIAS web interface following the ingest of data and the generation of school reports. Instructions for generating and using these reports are in section 3 of this document.

School Score Summary	

Purpose	• Displays for selected school all domains for all year levels:
	    ○ School Average
	    ○ Jurisdiction Average
	    ○ NationalAverage
	    ○ Top National 60%
	    ○ Bottom National 60%

What to expect	• Filters available for Year level • Can drill down on individual rows to view information graphically • Report can be downloaded as csv file

Student Domain Scores	

Purpose	• Displays for selected school scores for each student for each domain
	    ○ Raw score
	    ○ Scaled Score
	    ○ Scaled Score Std Error
	    ○ Domain Band
	    ○ Proficiency

What to expect	• Filters available for Year level and domain • Can drill down on individual rows to view information graphically • Report can be downloaded as csv file

Student Participation	
Purpose	• Displays participation codes for each student for each domain
What to expect	• Filters available for Year level • Can drill down on individual rows to view each domain test details as well as exemption, disruptions and DAC/PNP codes • Report can be downloaded as csv file

NAPLAN Codeframe	

Purpose	• Displays for each year level and domain:
	    ○ Subdomain
	    ○ Node
	    ○ Testlet Name
	    ○ Item Name

What to expect	• Can drill down on individual rows to view further information on Content, Writing Rubric and Scoring

NIAS Tools – Installation and User Guide

7.1 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

NIAS Tools – Installation and User Guide