NIAS - Support for Distributed Marking of Writing

Section 1 - NSIP Integration As a Service (NIAS)

1.1 What is NIAS?

NIAS is a suite of open-source tools designed to help users integrate and extract value from data based on the Australian SIF Data Model for Education. NIAS includes support for specific applications relating to NAPLAN, such as validation and manipulation of data contained in the Results and Reporting Dataset files (RRD) produced by the National Assessment Platform.

These notes describe the use of NIAS tools to extract data from the RRD to support the distributed marking of NAPLAN Online writing responses.

Section 2 - Installation

2.1 Pre-requisites

NIAS can be installed on Windows (32 or 64-bit), Linux (32 or 64-bit), or Macintosh. A browser is not required to support the writing extraction process using NIAS, as NIAS uses a command line interface. A browser will be required however to view the HTML writing scripts produced. On Windows machines, Firefox and Chrome are recommended. On Mac, Safari, Chrome and Firefox are recommended.

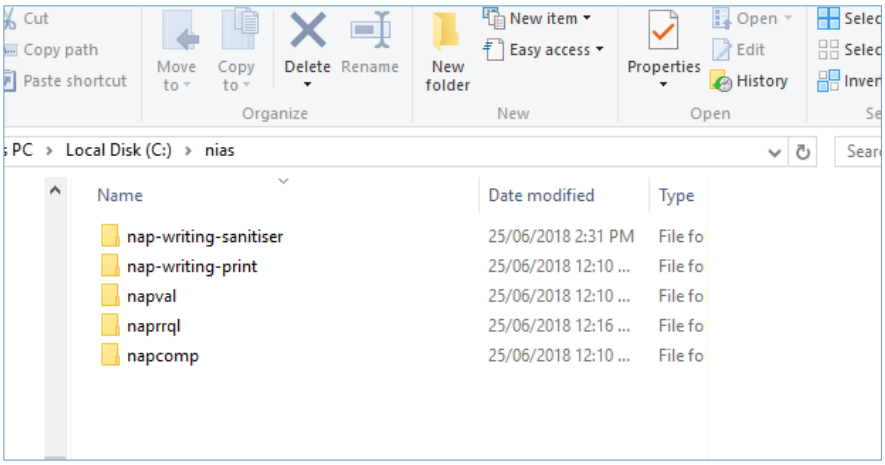
The NIAS toolset takes up approximately 40MB in hard disk space.

2.2 Installing files

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

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

3. Extract the contents of the zip file (e.g. Win64-1-0-0.zip) 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 creates 5 folders:

- nap-writing-sanitiser
- nap-writing-print
- napval
- naprrql
- napcomp

For further information about all components please refer to the *NIAS and NAPRR Install and User Guide* included in the download (under the naprrql folder).

Section 3 – Student participation/test attempt statuses and impacts on the RRD

It is important to note that RRD files generated during the test window are not finalised, and therefore their data representation is in transition.

When dealing with RRD files generated **prior to all schools being in the 'Results' phase**, users of the RRD should note that:

1. **Postponed attempts**: Students who have postponed test attempts may be included in the RRD files produced. These students will have a participation code of 'P'. Whilst policy and process strongly encourages all Writing test sessions to be completed in the first week, there is nothing systematically preventing a postponed test attempt from starting in Week 1, and being completed in Week 2.

This may result in a partial Writing test response being included in the RRD file and therefore being output via NIAS.

RRD users are **STRONGLY ENCOURAGED** to run the following report on the assessment platform to identity postponed test attempts:

- From the "Delivering" dashboard, under the Test Attempt Status chart, select "Postponed"
- The "Test Attempt" page will display, filter for Writing in the "Search for Assessment Event" filter.
- Export the results

- Students who are “Postponed” should be considered for removal from any Writing extracts produced by NIAS to ensure the final, completed responses are provided to external marking systems and not partial, incomplete responses.

2. **Altering test attempt status**: Functionality was introduced in 2019 allowing test attempts to be reopened by TAA level users. If Writing attempts are edited in this way these attempts should be evaluated by TAAs and, if necessary, included in subsequent writing extracts.

There is currently no report on the platform which provides details for test attempts which have been altered. TAAs will need to determine out of band which of these Writing responses should or should not be included in subsequent extracts.

3. **Open registrations**: If a student has not yet attempted a test, they are still included in the RRD file, and will have a participation code of ‘P’. A NAPStudentResponseSet response object will also be output for those students. The difference between the response object for open registrations, and for students who have actually attempted a test, is that the actual response script (the <Response> element) will not be present in the open registration instances of the response object, and no start time will be populated for the NAPEventStudentLink object. The <Response> element will be present in the response object for any attempted tests, even if it is empty or nilled. The NIAS Writing extract tool removes these open registrations (with no actual attempt) as part of its creation of the writing_extract.csv file.

In the finalised RRD generated at the conclusion of the test window, all open registrations will have been resolved: either their participation code will have changed to a value other than P, or else a response script will be supplied. This open registrations issue therefore does not impact RRD files generated once schools have performed clean-up activities and are in the ‘Results’ phase.

4. **Resitting tests**: If a test has to be readministered because of disruptions, as has been the case in 2019 NAPLAN, the approach currently taken is to reopen the existing test attempt for impacted students, rather than register them for a new test and risk having two different responses in the system for the same domain and the same student. This means that the previous test and the new test are not differentiated in the RRD: the identifier for the test that the student is registered for is *not* updated. (In fact in 2019, the test prompt for the readministered writing test is only circulated on paper.)

This means that TAAs have to record which student responses correspond to the original test, and which responses correspond to the readministered test. NIAS offers some assistance (including the date and time that the test was sat in reports, so that TAAs can infer which test attempts are for the new test.) But students can also sit the original test at any time during the examination period, for the reasons given above; so TAAs will have to put their own checks in place.

Section 4 – Writing Extract

The NIAS tool contains a component to extract all writing responses from the Results and Reporting data set, to support upload into existing writing marking systems.

The NIAS writing extract process generates a CSV file, writing_extract.csv, with one row per student, for all students registered for a writing test. It includes the participation status for the student, the year level, the ACARA ID for their school, the student’s local and TAA-assigned identifier, their Platform Student Identifier, and the contents of their writing script where available. It also includes

the word count of the writing script, the local identifier of the writing test, and the date and time when the test was administered.

The writing script output is in the same HTML format as in the Results and Reporting data set. The extract also includes one random anonymised identifier for each student record. This identifier, which is not contained in the Results and Reporting data set, can be used instead of the PSI when supplying writing scripts to external marking services, to preserve student anonymity.

The extract includes all writing responses for all year levels represented in the Results and Reporting data set; Year 3 will not be represented, as no registrations for Year 3 (offline) Writing are recorded in the National Assessment Platform.

The word count for the writing response ignores HTML markup, and counts contractions as two words (using the code library https://github.com/clipperhouse/jargon/tree/master/contractions). Hyphenated words are counted as multiple words. A token only counts as a word if it contains an ASCII letter (a to z) or a numeral.

4.1 Running the Writing Extract

1. Download the zipped Results and Reporting Dataset (RRD) file available from the NAPLAN Online Assessment Platform (one file for a given sector - NIAS should be provided one RRD file at a time) and unzip the file (**SIF.xml)** to the NIAS subfolder: **naprrql\in**

> **IMPORTANT:** NIAS comes bundled with a sample RRD file (**sample.xml.zip**) which includes sample student writing scripts. Move or delete the **sample.xml.zip file** from **naprrql\in** if you are using NIAS in production.

2. Navigate to the **\naprrql** folder in a console or terminal (in Windows type **cmd** in the Windows search box, then, if NIAS was extracted to C:\NIAS, type **cd\nias\naprrql**) and run:
   naprrql -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 **naprrql\in** folder.
3. Wait for the application to ingest the data. Progress will be displayed in the console. Note that the ingest process only needs to be done once unless the data changes (a Results Reporting data set with 50,000 students takes 6 minutes to ingest on a Quad Core i7 MacBook Pro).
4. **Ingestion complete** followed by **Datastore closed** indicates that the ingest process has completed. Don't close the console/terminal window.
5. Whilst still in the **naprrql** folder (e.g. C:\nias\naprrql) in a console or terminal, run:
   naprrql -writingextract on Windows, **naprrql --writingextract** on Linux and Macintosh.
6. Wait for the process to complete (the writing extract from a RRD with 50,000 students takes 2 minutes to generate on a Quad Core i7 MacBook Pro.)
7. Navigate to the **\naprrql\out\writing_extract** folder. Three files have been created: **writing_extract.csv, qaSchools.csv** and **qaPSI.csv**.
   - The writing_extract.csv report contains a row for every student in the source file that was registered for a writing test in the RRD, whether they have actually sat the test or not. Because the source RRD does not track paper tests, there will be no entries for Year 3 students. This is the file that the writing scripts will be generated from.

Test Year	Test level	Jurisdiction Id	ACARA ID	PSI	Local school student ID	TAA student ID	Participation Code	Item Response	Anonymised Id	Test Id	Word Count	Date	StartTime
2017	5	9	21213	R100000042R	819020519	819020519	P	This is a sample of 600 words \n\n <stron	i6ZzzZDHreMiRxLvnhT1x7	x00106808_Writing	600		
2017	5	9	21213	R100000029G	664875159	664875159	P	This is a sample of 600 words \n\n <stron	sKCZ60XSh1mZEi1kZkMr3R	x00106808_Writing	600		
2017	5	9	21213	R100000036S	476393126	476393126	P	This is a sample of 600 words \n\n <stron	LMkxeec9Bll3f2c26bm0kG	x00106808_Writing	600		
2017	5	9	21213	R100000049H	745864299	745864299	P	This is a sample of 600 words \n\n <stron	h1WssroVB8pZ4D0l0spiDA	x00106808_Writing	600		
2017	5	9	21213	R100000028M	746926784	746926784	P	This is a sample of 600 words \n\n <stron	4mz97lxRYmnuiQZILvKBLO	x00106808_Writing	600		
2017	5	9	21213	R100000035D	382707688	382707688	E		LmMpAx3o4Xm3tgwk653jo3	x00106809_Writing_alt	0		
2017	5	9	21213	R100000048G	417330656	417330656	S		48H58oB2Bzh6V0xEg917LV	x00106809_Writing_alt	0		
2017	5	9	21212	R100000011H	383655377	383655377	A		GFj4v0Xi4xhXjFOB5OCaTF	x00106809_Writing_alt	0		
2017	5	9	21212	R100000000K	812451070	812451070	P	This is a sample of 600 words \n\n <stron	pVmB0Mvg2fu1FUvAXJbpZ3	x00106809_Writing_alt	600		
2017	5	9	21212	R100000023R	624504082	624504082	P	This is a sample of 600 words \n\n <stron	rzWDRZBFhDVFm2XYwc6foM	x00106809_Writing_alt	600		
2017	5	9	21212	R100000001E	872347538	872347538	P	This is a sample of 600 words \n\n <stron	k8S7w701MrkJGH1GaIv2lO	x00106809_Writing_alt	600		
2017	5	9	21212	R100000009M	951674161	951674161	P	This is a sample of 600 words \n\n <stron	DKtX0EMHU0jH8Qj2BmgU1b	x00106809_Writing_alt	600		

   - qaSchools.csv contains a row per school with numbers of students against year levels, writing tests and participation codes. The counts of students registered for writing tests can be used to validate the contents of the writing_extract.csv report. Note: An additional set of columns per year level (Writing Extracts Y5/7/9) have been added which specify the number of script files actually generated by the NIAS tool. This will assist in reconciliation with external marking systems.

SchoolName	Sector	System	Independent	Type	Local ID	ACARA ID	State ID	District	Total Reg. Students for Writing	Students Reg. yr5	Students Reg. yr7	Students Reg. yr9	Students Reg. Test lvl5	Students Reg. Test lvl7	Students Reg. Test lvl9	Participated Yr 5 Writing	Participated Yr 7 Writing	Participated Yr 9 Writing	Not Enrolled	Absent	Cancelled	Exempt	Withdrawn	Sanctioned Abandon.	Refused
Altica Sec	Non-Government				x72860	21212			18	4	6	8	4	6	8	4	4	6				2		1	1
Bardish Se	Non-Government				x72861	21213			19	6	9	4	6	9	4	6	7	4					1		1
Dictyosipl	Non-Government				x72862	21214			19	11	5	3	11	5	3	11	5	3							
Dogmatic	Non-Government				x72863	21215			22	8	7	7	8	7	7	7	5	5				1	2		2
Empyreur	Non-Government				x72864	21216			18	4	6	8	4	6	8	4	6	6					1	1	
Entertain	Non-Government				x72865	21217			18	7	4	7	7	4	7	7	3	7					1		
Feeless Se	Non-Government				x72866	21218			21	5	11	5	5	11	5	4	8	4				2	1		2

- qaPsi.csv contains a list of PSIs generated in the writing extract. The file can later be used to exclude students with those PSIs, and only extract new students between 2 loads of the RRD file if required.

NOTE: If you wish to run a further validation to compare the csv file generated by the NIAS Writing Extract tool against the original source RRD file, NSIP has a tool available which runs on Linux or Mac. NSIP has used this tool in testing and can provide further information on request.

4.2 NAP Writing Sanitiser

NAP-writing-sanitiser is a standalone tool that sanitises the writing extract from NAPLAN student responses, to strip extraneous HTML styling introduced through copy-paste by students during testing, and to wrap responses in HTML where responses lack any HTML markup. Such sanitisation becomes necessary because responses with such copy-pasted HTML may be illegible when rendered as writing scripts in the next step: the writing prompt introduced into the HTML may end up covering over the actual response.

The NAP-writing-sanitiser tool is designed to work with the .csv file that is output by the naprrql data analysis tool when naprrql is run with the --writingextract flag. The use of the NAP-writing-sanitiser is optional.

1. To run, the steps for data ingest and writing extract must have already been run. See instructions in section 3.1. (steps 1-6)
2. Navigate to the **\naprrql\out\writing_extract** folder. Copy the file **writing_extract.csv** to the folder **\nap-writing-sanitiser\in**.
3. Navigate to the **\nap-writing-sanitiser** folder in a console or terminal (in Windows type **cmd** in the Windows search box, then, if NIAS was extracted to C:\NIAS, type **cd \nias\nap-writing-sanitiser**) and run: **nap-writing-sanitiser**. The nap-writing-sanitiser tool will then generate two files:
   - **out/writing_extract_sanitised.csv**: a file with the same structure as writing_extract.csv, but with the HTML content sanitised
   - **out/sanitiser_report.csv**: a before-and-after list of all sanitised responses

The following HTML markup in the responses is preserved during the sanitation process:

- The elements "strong", "em", "span", "p", "ol", "ul", "li", "br", "u", "font", "h1", "h2", "h3", "h4", "h5", "h6"
- The attribute "size" on the element "font", with a numeric value.
- The following values of the attribute "style" on any element, singly or in combination:
  - text-decoration:underline;
  - text-decoration-line:underline;
  - font-size:16px;
  - font-size:18px;
  - font-size:large;
  - text-align:left;
  - text-align:center;
  - text-align:start;
  - background-color:rgba(255, 255, 255, 0);

All other attributes are stripped, including CSS classes, and other values of the "style" attribute. All other HTML elements are stripped; this includes the bulk of elements in the Writing prompt. As a result, the HTML styling specific to the writing platform environment, which can cause difficulties outside of that environment, is removed; only styling introduced through deliberate use of the platform editor menus (such as boldface, text alignment, and lists) is preserved.

If any responses lack any HTML markup, the tool inserts a <p> wrapper around the response, and in any instances of double carriage return. This ensures that all responses are rendered consistently as HTML, whether they have HTML markup in the RRD or not.

To make sense of the sanitise_report.csv report, we suggest a visual diff tool, that highlights the changes to markup introduced by the sanitiser. https://text-compare.com is one example of such a tool.

4.3 NAP Writing Print Tool

The nap-writing-print tool allows generation of a separate HTML file for each student's writing response. These files can be redistributed to support manual marking of writing scripts if needed.

The nap-writing-print tool is included in the standard install (as per Section 2 above).

IMPORTANT: NIAS comes bundled with a sample writing extract file (**sample_writing_extract.csv**) which includes sample student writing scripts. Move or delete the **sample_writing_extract.csv file** from **\nap-writing-print\in** if you are using NIAS in production.

WARNING: the writing tool code is hardcoded to the headers of the writing_extract file. Users should refrain from renaming the headers of the writing_extract file (i.e. don't edit the /app/naprrql/reporting_templates/writing_extract/itemWritingPrinting_map.csv columns).

This section assumes that the earlier steps (to create the writing_extract.csv) in section 3.1 and the optional steps to run the NAP-writing-sanitiser in section 3.2 of this document have already been completed.

To generate the writing scripts (in HTML format):

1. If you have used the Nap-writing-sanitiser tool, copy the file **writing_extract_sanitised.csv** from **\nap-writing-sanitiser\out** to the **\nap-writing-print\in** folder. If you have not used the sanitiser tool, copy the file **writing_extract.csv** from **naprrql\out\writing_extract** to the **\nap-writing-print\in** folder.
2. Navigate to the **\nap-writing-print** folder (e.g. at the command prompt type cd \nias\nap-writing-print) in a console or terminal and run **nap-writing-print** on Windows, Linux and Macintosh.
3. The nap-writing-print tool will then create individual HTML files for each writing response that it finds in the **writing_extract.csv** file, under the structure outlined below.
4. A number of files will be generated in the **\nap-writing-print\out** and **backup** folders.
5. The **backup** folder contains backup/s of the input CSV file, with a separate date-stamped folder for each time the nap-writing-print tool is run. This allows regeneration of HTML files from the source writing_extract.csv file (with the same anonymous student identifiers).
6. The **out** folder will contain a **schools** folder, outputting HTML scripts for each student organised by school (ACARA Id), and a **yr-level** folder, outputting HTML scripts for each student organised by year level.

7. Each numbered folder within the **school** folder and **yr-level** folder contains an **audit** folder and a **script** folder.
8. The **script** folder contains a HTML file per student. The file name for each HTML file is N_X_ID.html where N is the identifier for the jurisdiction (state or territory), X is the student's participation status, and ID is the anonymised ID of the student. The HTML internally consists of the anonymised ID for the student, followed by the response (or a notice that there was no response).
9. The **audit** folder contains a HTML file per student, with the same naming convention; its contents are the full information known about that writing script from the input CSV file, *including the student's PSI*. This means scripts can be distributed independently, but manually reconciled if needed.

4.4 Optional - Creating writing extracts for only new student responses

In 2019, a number of RRDs will be generated for Test Administration Authorities (or equivalents) during the test window to speed up the marking of writing responses. Each RRD generated from the platform is cumulative.

For example:

- Extract 1 taken after week 1 of testing contains responses from Students A, B and C
- Extract 2 taken after week 2 of testing contains responses from Students A, B, C, D and E
- Extract 3 taken after week 3 of testing contains responses from Students A, B, C, D, E and F

As discussed in Section 3.3, the extract for each week will contain both a registration object and a response object for every student; but the response objects for students who have not yet attempted the test will not yet be populated, and are identified and ignored by the NIAS writing extract process.

To facilitate the loading of student writing responses into external marking systems, by identifying only the additional students with actual responses after each load, new functionality has been added to NIAS. This will support:

- Identification of students A,B, C in week 1
- Identification of students D and E in week 2
- Identification of student F in week 3

If you optionally wish to generate another extract without including the students already generated in previous RRDs:

- Complete processing the first RRD file (ie week 1) as per instructions in section 3.1 above.
- Once you have run the extract the first time the **qaPsi.csv** file will have been generated in the C:\nias\naprrql\out\writing_extract folder. This contains a list of all students in the first extract whose registrations are not still open: that is, all students with a Participation Code of P and a recorded response, and all students with a Participation Code other than P. (In the example above, the response objects for students D, E, F will not contain a script, and are ignored.) **Move** this file to the **C:\nias\naprrql** folder, and (to avoid confusion) rename it as **filter.csv**. It will be used to indicate the students to skip in the next iteration.
- IMPORTANT – The first writing_extract.csv and qaSchools.csv files will be overwritten if NIAS is run a second time. If you wish to keep these files, **Move** or **Rename** the files **writing_extract.csv** and **qaSchools.csv** that exist in the **C:\nias\naprrql\out\writing_extract** folder.

- Obtain the second RRD file (ie week 2). Whilst still in the naprrql folder (e.g. C:\nias\naprrql) in a console or terminal, run:
  naprrql –writingextract –psiexceptions filter.csv on Windows,
  naprrql --writingextract -- psiexceptions filter.csv on Linux and Macintosh.
  (where filter.csv is the filename containing the PSI of students to skip in the current iteration—in this case, the students included in the first extract (week 1). If you have renamed the file, use the new filename).
- This will create new copies of **writing_extract.csv, qaSchools.csv** and **qaPsi.csv** (with different contents compared to week 1, i.e. the NAPStudentResponseSet objects will contain writing scripts or resolved registrations for students D and E, but not yet for student F.)

- Of these files, writing_extract.csv contains only the students added from the previous iteration (the students not already given in filter.csv, but that do now have either a response script, or a participation status other than P). The new qaPSI.csv file generated lists only those newly added students. On the other hand, the qaSchools.csv reports the counts of all students present in RRD file, old and new. That is because the qaSchools.csv is used to reconcile the obtained RRD file against the known registrations for Writing tests in each school: TAAs will know there are no more extracts to wait on when the counts of the two match.
- If you wish to run extraction a third time, repeat the steps given above. The new extract will need to skip both the students extracted in the first iteration, and the students extracted in the second; for that reason, the qaPSI.csv file obtained for the second iteration will need to be appended to the qaPSI.csv file obtained for the first as a new filter.csv file, as the list of students to ignore passed through the **–psiexceptions** flag.
- Should you need to edit the list of students to skip in a future iteration, do so by adding or removing PSIs in the **filter.csv** file. For example, if you remove the PSI for student A from the file, then both the first and the second iteration will contain the writing extract for that student.

4.5 Optional - Creating writing extracts for a defined set of students

- By the third week, it may be more useful for TAAs to generate writing extracts according to a whitelist of which students to include, rather than a blacklist of which students to exclude. That will be particularly useful for dealing with students resitting the test, and for the final reconciliation (described below).

- This mode is also supported in NIAS, through the command
  naprrql –writingextract –psiwhitelist whitelist.csv on Windows,
  naprrql --writingextract -- psiwhitelist whitelist.csv on Linux and Macintosh.

  The **whitelist.csv** file is a CSV file, with no header, and contains two fields: the PSI of the student to include in the extract, and (optionally) the writing test identifier to use for that student. The second field is included so that readministered tests can be differentiated from original tests for markers, since the two tests are not currently differentiated in the RRD. If the second field is not included in a row of the CSV file, the original writing test identifier will be left alone in the writing extract.

4.6 Final clean-up of Writing responses for 2019

Given the issues posed with the interim use of the RRD to support the marking of writing (postponed student test attempts, and the ability to reopen test attempts and to change participation status), users of the RRD for the purposes of marking writing responses are advised to **perform a final reconciliation once all schools have entered the 'Results' phase,** to ensure all student writing attempts have been extracted correctly and faithfully represent the student's final writing response.

If using the steps detailed in section 4.4 (to only extract new responses) particular attention should be paid to student attempts which indicated some form of non-attempt (eg 'R') which may have been changed at a later date and would therefore be ignored in subsequent extracts.

As a result of that exercise, TAAs should have a list of PSIs for students for which a test script still needs to be generated for marking. TAAs should use the whitelist capability of NIAS, described above, to do so.

Section 5 – Students with multiple test status or writing scripts in the RRD

The Interim RRD files generated during the test window are not finalised, and therefore their data representation is in transition.

The NIAS **csvdiff** functionality is intended to compare between CSV reports generated by two different RRD files and report the differences for investigation by TAAs. After an initial RRD file has been ingested and reports have been generated, subsequent RRD file/s can be ingested and the output compared to the previous file.

1. **Test attempt status**: If test attempts have changed between RRD data extracts, these will be identified.

2. **Resitting tests**: If a test has to be readministered because of disruptions, as has been the case in 2019 NAPLAN, the approach currently taken is to reopen the existing test attempt for impacted students, rather than register them for a new test and risk having two different responses in the system for the same domain and the same student. Students who have retaken tests may be included in the RRD files produced with a different test attempt status and different test response in a later RRD file. The comparison will identify the difference in the number of words in the test response. This means that TAAs will be able to identify which student responses correspond to the original test, and which responses correspond to the readministered test.

This section assumes that the NIAS tools have been installed locally. If a previous data ingest has already occurred and comparison reports have been generated, these must be copied elsewhere if they are required, because they will be overwritten by the new data. Begin at **section 5.2** as you have already completed section 5.1 when the initial RRD was loaded.

5.1 Consuming the initial RRD load

1. Download the zipped Results and Reporting Dataset (RRD) file available from the NAPLAN Online Assessment Platform (one file for a given sector - NIAS should be provided one RRD file at a time) and unzip the file (**SIF.xml)** to the NIAS subfolder: **naprrql\in**

IMPORTANT: NIAS comes bundled with a sample RRD file (**sample.xml.zip**) which includes sample student writing scripts. Move or delete the **sample.xml.zip file** from **naprrql\in** if you are using NIAS in production.

2. Navigate to the **\naprrql** folder in a console or terminal (in Windows type **cmd** in the Windows search box, then, if NIAS was extracted to C:\NIAS, type **cd\nias\naprrql**) and run:
   naprrql -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 **naprrql\in** folder.
3. Wait for the application to ingest the data. Progress will be displayed in the console. Note that the ingest process needs to be done each time the data changes (a Results Reporting data set with 50,000 students takes 6 minutes to ingest on a Quad Core i7 MacBook Pro).
4. **Ingestion complete** followed by **Datastore closed** indicates that the ingest process has completed. Don't close the console/terminal window.
5. Whilst still in the **naprrql** folder (e.g. C:\nias\naprrql) in a console or terminal, run:
   naprrql -report on Windows, **naprrql --report** on Linux and Macintosh.
6. Wait for the process to complete
7. The report process will create **\napprql\out** and two sub folders **school_reports** and **system_reports**. The **school_reports** folder will contain a separate folder for each school in the jurisdiction.
8. Reports are ready for analysis.

5.2 Consuming a subsequent RRD load

8. Move the **\school_reports** folder from the **\out** folder to the **naprrql** folder and rename it to **dir1_reports** so the relevant files are not overwritten. (Note the name is not important, but you will need to use the name in the command line to run the comparison.)
9. For each of the school subfolders in the **dir1_reports** folder, retain only the two reports prefixed with 'compare' in the file name. Discard (or move) all the others. This will minimise the time it will take to run the comparison tool. (The other reports have not been tailored for the purposes of cohort comparison, so if you leave them in the folders, the comparison will report discrepancies between the two RRDs that are not significant.)
10. To consume another RRD load, follow steps 1 -8 above to ingest and re-generate reports. This will generate a new **\out** folder. As always, rename any folders you are going to be comparing, so they do not get overwritten.

5.3 Comparing two RRD loads

11. In the **naprrql\out\school_reports** folder (containing the newly generated reports), for each school subfolder, retain only the two reports in the prefixed with 'compare' in the file name. Discard (or move) all the others.
12. Navigate to the **\naprrql** folder in a console or terminal (in Windows type cmd in the Windows search box, then, if NIAS was extracted to C:\NIAS, type cd\nias\naprrql) and run:
    naprrql –csvdiff dir1_reports out\school_reports on Windows
    naprrql -- csvdiff dir1_reports out\school_reports on Linux and Macintosh
    NOTE: if you named your folder something other than dir1_reports, replace dir1_reports with the name you used.

    The compare function recursively goes through every .csv file in the first folder and looks for a matching file in the second folder; if one is found it compares the two.

If there is not a matching file in the second folder, an error is reported. If there is a file in the second folder that is not in the first folder, it is ignored.

The two different folders are presumed to contain CSV files generated by NIAS: any CSV files are not parsed to ensure their columns are in the same order (because NIAS always outputs columns in the same order), but any two files with the same name are sorted by row before NIAS compares them.

13. A new folder **\compare** will have been created under the naprrql folder. The results of the comparison is output to a file in the compare folder, with its name timestamped. For each pair of CSV files where a difference is found, the output prefixes with - any lines removed in the second folder; it prefixes with + any lines added in the second folder, and it puts any changed lines next to each other in the report, with - prefixing the line in the first folder, and + the line in the second folder. For example:

```
Comparison, out/system_reports/actSystemDomainScores.csv,
out1/system_reports/actSystemDomainScores.csv

-21213,Hippocentaur Primary
College,Williams,Janet,R100000042R,5,5,Reading,,,,A1:D1:F0,S
-21213,Hippocentaur Primary
College,Williams,Janet,R100000042R,5,5,Writing,8,1.00,575.00,,P
+21213,Hippocentaur Primary
College,Williams,Janet,R100000042R,5,5,Writing,8,1.00,575.00,,G
```

This means that the Reading results for Janet Williams have gone missing in the out1 folder: there is no corresponding line in the ACT System Domain Scores report there. It also means that the Participation Code for the Writing results for Janet Williams has changed between the two folders, from P in the out folder, to G in the out1 folder.

Summary of comparison results:

For each pair of CSV files where a difference is found

-	Line has been removed from the second folder but existed in the first
+	Line added in second folder which did not exist in the first folder

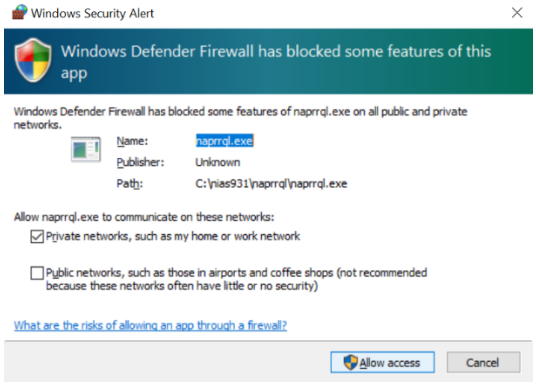
Changed lines are next to each other in the report

-	The line in the first folder is the original data
+	The line in the second folder is the changed data

Additional Notes:

- Performance: NIAS runs best on machines with a SSD drive and at least an i5 processor and 8GB of RAM. Current performance on an i5 MacBook creates 200,000 html files (4 files for each of 50k students) in around 1 minute.
- As the html output is constructed entirely from the contents of the input file (**writing_extract.csv)**, for safety at the end of a run a timestamped folder is created in the **/backup** folder of the **nap-writing-print** folder so that the same html files can be generated at any time in the future, even if the working .csv file has been over-written.

- The output HTML script files maintain the paragraphing of the original input from the user: bold text, underlined text, italic text, and ordered and unordered lists are supported as provided via the online NAPLAN editor component.
- As the writing extract tool will generate a lot of files, it is best run on 64-bit environments where constraints on the number of files in a directory or folder are not an issue if large input files are being processed.
- Virus scanners & firewall requests: On Windows, NIAS may request network access on the local machine (as seen via a Windows firewall request). Note that NIAS does not access outside networks, websites, or the assessment platform directly. Internal NIAS components transfer information locally, internal to the current machine only and requires this internal access to function correctly.



Further support:

For additional NIAS support please contact NSIP (info@nsip.edu.au) or call 03 9910 9827

Appendix A: Contents of file writing_extract.csv

From NIAS version 1.01+ , the following fields will be included in the writing_extract.csv file:

i. Test Year (e.g. 2018)
ii. Test Level (e.g. 5,7,9)
iii. Jurisdiction ID (e.g. 2 = Victoria)
iv. ACARA ID
v. PSI
vi. Local school student ID
vii. TAA student ID
viii. Participation code
ix. Item Response (the student's response in HTML text)
x. Anonymised ID (random alphanumeric GUID mix of numbers, upper/lower case letters)
xi. Test ID (the local test ID for the writing test e.g. x00115999)
xii. Word Count