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.

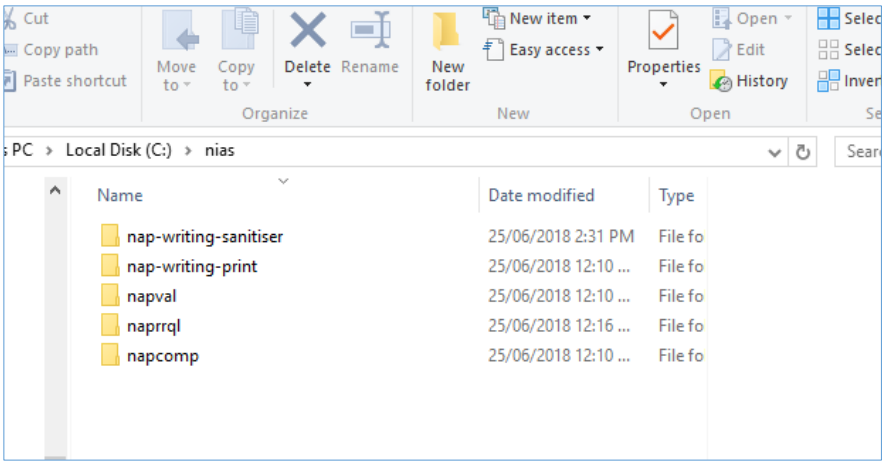
The NIAS toolset takes up approximately 40MB in hard disk space.

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 – 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. 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.

Word counts for each of the student writing responses are included in the writing_extract.csv file. The word count ignores HTML markup, and counts contractions as two words. 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.

3.1 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
2017	9	9	21212	R100000011H	46299451	46299451	P	<p>This is a sampl	s4X54jMsXQJGjr3MF	x00106802	300
2017	9	9	21212	R100000019K	876603939	876603939	R		4oA3yWpguuEVvVq	x00106802	0
2017	9	9	21212	R100000016D	767981676	767981676	E		xV7Z7XMu7871KdO6	x00106802	0
2017	9	9	21212	R100000002D	362156703	362156703	P	<p>This is a sampl	ja21FY66uwuYMhQ3	x00106802	300
2017	9	9	21212	R100000012P	428224238	428224238	P	<p>This is a sampl	ce4X6qa68PN1v2P6t	x00106802	300
2017	7	9	21212	R100000003S	442580663	442580663	S	<p>This is a sampl	GMf3MZkgizthrXikm	x00106801	604
2017	7	9	21212	R100000004R	557839832	557839832	P	<p>This is a sampl	0uvPiSL88BEGHsSPN	x00106801	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.

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.

3.2 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.

3.3 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.

3.4 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

A further complication is that, because the RRD files received during the test window are not finalised, their data representation is in transition. In particular, response objects (NAPStudentResponseSet) will be generated for students who are registered for a test, but have not yet attempted it ("open registrations"); and the participation code for those registrations (under NAPEventStudentLink) will be given in the RRD as P. The difference is that the actual response script will not be populated in the RRD file.

So in the example above, in Extract 1, there will be a NAPEventStudentLink object generated for students D, E, F, with a participation code of P. In fact there will also be a NAPStudentResponseSet object generated for those students. But unlike students A, B, C, the NAPStudentResponseSet will not contain an ItemResponse element, containing the students' actual response, because that response is not yet available.

Any such open scripts will be detected in the generation of the writing extract, and *ignored*. In the finalised RRD generation 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 will be supplied.

To facilitate the loading of student writing responses into external marking systems by identifying only the additional students 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. 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 qaPsi.csv on Windows,
  naprrql --writingextract -- psiexceptions qaPsi.csv on Linux and Macintosh.
  (where qaPsi.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 qaPSI.csv, but that do now have either a response script, or a participation status other than P). The new qaPSI.csv file 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 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 to the copy of qaPsi in the **C:\nias\naprrql** directory. 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.

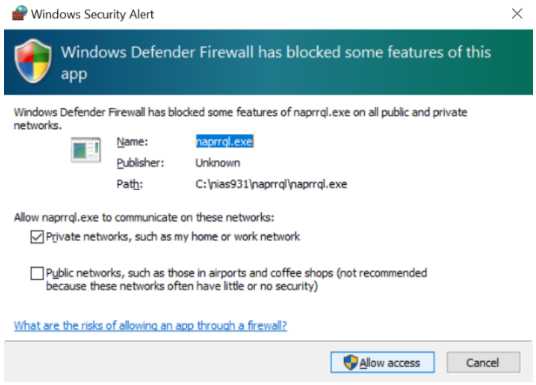
WARNING: This process assumes that once a registration is resolved, it does not need to be output again; any students marked as R in the first extract, for example, can be ignored in subsequent runs. That is not the case, however, either principals or TAAs can change the participation status of students during the test window, and test attempts can be postponed and resumed. For that reason, TAAs should run a reconciliation at the end of the test window, between the first extract received, and the final extract received.

- From the **writing_extract.csv** file generated in Week 1, extract all students whose participation status is other than P.
- Identify the same students in the **writing_extract.csv** file generated from the final extract.

- If any of those students have changed status to P, you will need to generate writing scripts for those students as well, by manually removing their PSIs from the **qaPsi.csv** file you have been maintaining to exclude students.
- You may need to do the same for any students in the Week 2, Week 3 etc extracts with participation status other than P, whose status has since changed to P.

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