WORD ASSOCIATION DATA PROCESSOR 1.0.1
THE MANUAL

Andreas Buerki
buerkiA@cardiff.ac.uk

CONTENTS

1 INTRODUCTION

The Word Association Data Processor (WADP) is open-source software that automates key aspects of the processing of large amounts of word association data gathered from respondents in word association tests. Word association data are typically gathered by asking respondents to write down the first word (or the first few words) that come to mind given a cue word. The word(s) that are written down in response to cues, are referred to as *responses*. Figure 1 shows examples of cue-response pairs.

Responses are then typically analysed by looking for patterns in word associations across or within individuals, or across or within cues. One way to analyse associations is to categorise the relationship between cue and response. At a rudimentary level, the first cue - response pair in figure 1 could be categorised as instantiating a *paradigmatic* relationship between cue and response (e.g. *boy* and *girl* are alternatives that can be used in the same slot in a sentence). The second cue - response pair could be categorised as instantiating a *syntagmatic* relationship since a *house guest* is a complex term, the parts of which can occur together in the same sentence, next to each other. A more elaborate categorisation of word associations is presented in Fitzpatrick et. al. (2015:18-9). Another way in which cue - response pairs can be analysed is by looking at typical or *primary* responses to cues and how typical individual respondents' responses are.

boy ⇒ girl
house ⇒ guest

Figure 1: Two cue - response pairs

The WADP is designed to help researchers working with large amounts of word-association data to process cue - response pairs efficiently and consistently with regard to both categorisations as well as deriving primary responses. This manual introduces the reader to the features of the WADP and how it may be used. A video tutorial covering the basics is found at https://vimeo.com/603190447.

The manual is structured as follows: the next section lists the key features of the WADP. This is followed by a section with installation information. An introduction to basic operations is presented in section 4. More advanced features and their operation are introduced in section 5 and this is followed by a section featuring FAQs (section 6). The guide concludes with information on software licensing and copyright and an appendix describing file formats.

2 KEY FEATURES

The goal of the WADP is to facilitate the efficient and consistent categorisation of responses to word association tasks. The WADP comprises of three modules with the following key features:

Categoriser module:

- a tidy and efficient interface for the manual categorisation (i.e. rating) of cue - response pairs
- automatic categorisation of responses in cases where categorisations for the relevant cue - response pairs are found in a database of past category assignments
- automatic storage of all new categorisations in the database
- tracking of respondent IDs and rater IDs (if provided) in all in- and output files.
- tracking of rater IDs in categorisations stored in database files

Reporter module:

- automatic creation of individual response profiles
- automatic creation of cue profiles
- automatic creation of primary response lists
- display of inter-rater agreement ratios where several raters categorised the same data

Administrator module:

- automatic comparison and reporting of differences between database files
- support for inter-rater agreement procedures through interactive resolution of differences between two database files
- automatic combination of two database files
- turning a fully-categorised output file into a database file

3 COMPATIBLE OPERATING SYSTEMS AND INSTALLATION

Compatible Operating Systems

The WADP is written in the bash shell scripting language and as such can be run on wide range of operating systems, including Linux, macOS and Windows 10, if the Windows Subsystem for Linux (WSL) is installed. All recent versions of these systems include bash version 3.2 or later which is minimally required to run the WADP. The WADP was tested and confirmed working on macOS 11, Ubuntu 20.04 LTS and WSL on Windows 10.

Installation under Linux

Please run the `install.sh` script from the terminal. Should the installer fail, a manual installation can be carried out by placing the `WADP.sh` script somewhere convenient (preferably in a location that is in the user's `$PATH` variable). A Linux-specific installation video is available at vimeo.com/603098792. If installed via `install.sh`, a desktop shortcut (yellow icon) should appear on the desktop. The WADP is launched by double-clicking on this, or by typing `WADP.sh` into a terminal window. To test the installation, please follow the steps outlined in the README file.

Installation under Apple's macOS

A macOS-specific installation video is available at vimeo.com/603176910. Double-click on the `macOS_installer` inside the `WADP` folder, give macOS the necessary permissions. If a message appears saying that it cannot be opened because it is from an unidentified developer, click OK and ctrl-click/right-click on the script `macOS_installer` , then choose 'open' from the menu that appears, then click 'Open'. A Terminal window will pop up. If it complains about 'permission denied', follow these steps:

- open a new tab in the Terminal window (press 'command-T' or choose Shell > New Tab)
- type chmod +a followed by a space into the Terminal window
- drop the install.sh script (not `macOS_installer`) into the same Terminal window and press ENTER.
- now double-click on the `macOS_installer` script again and follow the instructions displayed.

A yellow icon should appear on the desktop (if selected) and in the Applications folder. The WADP is launched by double-clicking on this icon. The icon can be placed anywhere. Alternatively, the `WADP.sh` file can be renamed `WADP.command` and made executable and placed anywhere convenient, such as the application folder. To test the installation, please follow the steps outlined in the README file.

```
WORD ASSOCIATION DATA PROCESSOR
 version 1.0

 Please choose a module and press ENTER:

 (c) categoriser
 (r) reporter
 (a) administrator
 (x) exit
```

Figure 2: The WADP main menu

Installation under Windows

Under Windows, before the WADP is installed, the Windows Subsystem for Linux needs to be activated and the Microsoft Store's Ubuntu version installed. Instructions for how to do this can be found here and an installation walk-through for Windows is available at vimeo.com/603102292. Once installation is complete, a yellow icon should appear on the desktop. The WADP is launched by double-clicking on this icon, or by typing `WADP.sh` into a terminal window. To test the installation, please follow the steps outlined in the README file.

4 HOW TO USE THE WADP

This section describes the basic functionality of the WADP. A video tutorials is additionally available at vimeo.com/603190447. The WADP is started by double-clicking on its icon. The main menu as shown in figure 2 appears. From here, the different modules of the WADP are accessible. The main menu is automatically returned to after a task has completed.

The Categoriser module

This module helps categorise word association data. Figure 3 shows a schematic summary of the input and output files of the categoriser module. The only required input file is a .csv file containing the word association data to be categorised. This file needs to be formatted according to the formatting specification for in-files given in the appendix (page 14). The `test_data` directory contains two in-files of this format named `five.csv` and `five-extra.csv`. A file of this type might be constructed using a spreadsheet application (such as Excel or Open Office) and exported as a csv file, making certain that commas are used as field separator. Another common source of such data are online survey sites like onlinesurveys.ac.uk. Optionally, a database file of previous ratings can be supplied as well. These files are generated automatically by the WADP every time the categoriser

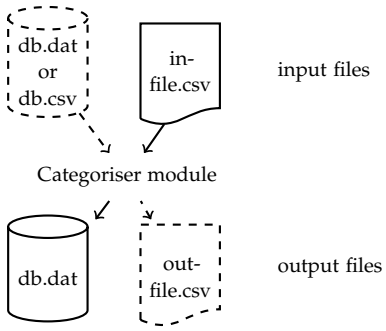

Figure 3: In- and output files of the categoriser module.
Note. Dashed files are optional

module is run (this is the `db.dat` output file of figure 3). If a database file is supplied in the input, any cue - response pairs for which a rating exists in the database will be categorised automatically rather than presented for manual categorisation. The appendix contains information of the file format of WADP-database files. It is a plain text format and can be examined in a plain text editor, though there is usually is no need to view or edit a database file manually.[1] The `test_data` directory contains two database files of this format named `example-database.dat` and `example-database2.dat`.

Once input files are received, the categoriser module will present cue - response pairs for categorisation. The full rating screen as shown in figure 4. By default, it is assumed that categories according to Fitzpatrick et. al. (2015:18-9) will be assigned. Additional categories or an entirely different set of categories can be specified if desired (see section 5). Note that it does not matter whether lower or upper case letters are entered. If no rating is entered (i.e. if merely `ENTER` is pressed), this is accepted as a valid response and will appear as an empty field in the output file. However, if that same pair is to be categorised again in the future, the programme will ask for a categorisation again, treating it as though no categorisation had ever been given.

Once categorisation of the input data is complete (or the categorisation process is interrupted), the categoriser produces a new (or updated) database file and a .csv output file that corresponds to the .csv input file with added columns for the category of cue - response association and a rater ID. If the categorisation process is exited before it is complete, out output .csv file is produced only if specifically requested.

The Reporter module

A summary of input and output of the reporter module is shown in figure 5. The module takes as input a categorised .csv file as produced as output by the categoriser module.[2] The `test_data` directory contains a csv-file of this format named `categorised-five-extra.csv`. From this input file, the module produces up to four types of reports as listed below. Apart

[1] See section 5 for more information on how to edit a database file.
[2] For a primary responses report, a non-categorised csv input file (of the same format as the input csv file to the categoriser module) is alternatively also acceptable.

```
Please categorise the following pair:

              irony -> IRONIC

enter a choice and press ENTER:

 (A)     Affix manipulation                          irony -> ironic
 (CR)    cue - response collocation                  fence -> post
 (CRRC)  cue - response & Response Cue collocation   rock -> hard
 (E)     Erratic                                     wolf -> and
 (F)     similar in Form only                        fence -> hence
 (I)     two-step association                        weak -> Monday, via week
 (L)     Lexical set                                 bean -> vegetable / pea
 (LCR)   Lexical set & cue - response collocation    gold -> silver
 (LRC)   Lexical set & Response-Cue collocation      cheese -> bread
 (OC)    Other Conceptual                            fence -> field
 (OCCR)  Other Conceptual & cue - response colloc.   long -> corridor
 (OCRC)  Other Conceptual & Response-Cue colloc.     attack -> knife
 (RC)    Response-Cue collocation                    fence -> electric
 (S)     Synonym                                     delay -> impede
 (SCR)   Synonym & cue - response collocation        torch -> light
 (SRC)   Synonym & Response-Cue collocation          shove -> push
 (SS)    Synonym in wider sense (not necessarily     joint -> unification
         same part of speech or number)

 (X)     exit (work will be saved)
```

Figure 4: The rating screen

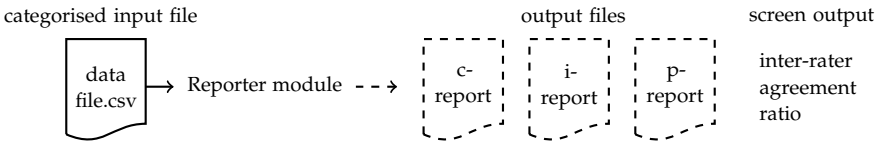

Figure 5: In- and output files of the reporter module

from the inter-rater agreement report which is only displayed on-screen, the other reports produce an output report file in csv format.[3]

INDIVIDUAL RESPONSE PROFILES (I-REPORT): lists instances of each cue-response association category summed *by respondent*. A schematic example is shown in table 1. Respondent IDs are only listed if present in the input file.

respondent ID	CAT1	CAT2	CAT3
respondent 1	3	21	1
respondent 2	0	16	33
respondent 3	24	5	10

Table 1: Individual response profiles

CUE PROFILES (C-REPORT): lists instances of each association category summed *by cue*. A schematic example is shown in table 2.

cues	CAT1	CAT2	CAT3
cue 1	89	234	1
cue 2	143	259	0
cue 3	6	272	478

Table 2: Cue profiles

PRIMARY RESPONSE LISTS (P-REPORT): csv-files that list all responses to each cue in order of frequency. Typically the most frequent response for each cue or the few most frequent ones are considered primary responses. An example list is shown in table 3, where the first line after the header would show the (most) primary responses.

blue	frequency	fence	frequency	irony	frequency
WATER	3	POST	4	IRONIC	2
GRASS	1	COW	1	TWISTED	2
CLOUD	1			SHARP	1

Table 3: Primary responses list

[3] Note that where applicable, reports list categories in alphabetical order and only categories to which assignments were made appear in reports.

INTER-RATER AGREEMENT REPORT If categorised word-association data are fed in that were derived from a categorisation database that had inter-rater resolution mark-up (for an explanation of this, see the description of the administrator module, below), then a report can be displayed on-screen which gives the ratio of inter-rater agreement, that is, the number of categories where both (or all) raters agreed on categorisation, divided by the total number number of categories that appear in the data file. For databases to be marked up with agreement, they need to have been created with WADP v. 0.6 or later. An example of an inter-rater agreement report is shown in figure 6.

```
============================================================
Inter-rater agreement report for categorised_seven.csv
============================================================
total ratings in file:  26
ratings marked as having needed resolution:  9
inter-rater agreement ratio:  .6539
============================================================
The accuracy of the report depends on an agreement-marked
database having been used to produce the data file.  This will be
the case if the database was created using WADP v.  0.6 or later.
Press ENTER to continue.
```

Figure 6: Example of an inter-rater agreement report

The Administrator module

This module performs a number of administrative tasks in relation to database files. It has four functions available as shown in figure 7. The first two of these functions can be useful, for example, when wishing to establish inter-rater agreement for a data set that was categorised (i.e. had categories assigned) by two different raters. One may first wish to simply produce a report listing any differences, and later, when one is certain of the correct category assignment for each instance where raters differ, one can then use the resolution function to resolve differences (the programme will ask which categorisation should be

```
What would you like to do?
(R) resolve differences between 2 database files now
(P) produce a list of differences only
(C) combine two database files into one
(T) turn categorised csv file into a database
```

Figure 7: The administrator menu

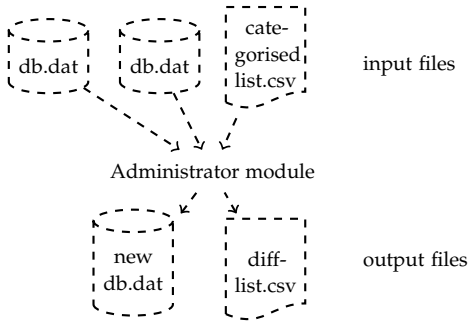

Figure 8: In- and output files of the administrator module.
Note. Dashed output files depend on options selected; diff-list is a list of differences.

used for all instances where categorisations differ) and end up with an agreed database file that can be used to assign categories to the same or new input files. The database will also contain a mark-up for each categorisation where raters initially disagreed. This feature is used to derive inter-rater agreement in the reporter module. Refer to the FAQs (p. 11) for a suggested inter-rater agreement procedure.

The (C)-function simply combines two database files automatically. While this generally assumes that there are no conflicting categorisations in the two databases, the software carefully checks for conflicts and resolves them by giving priority to the first database file. Any conflicts will also be logged and can then be reviewed by consulting the logfile produced.

The (T)-function turns a categorised output file (such as those produced by the categoriser module) into a database file of the .dat format.

Functions (R), (P) and (C) require as input two database files and produce a combined database file.[4] Optionally, a log file can be created that details the differences and decisions made to resolve them. Figure 8 shows a schematic overview of input and outfiles of the administrator module. Regarding naming conventions of database files, also see the box of important notes on database files on page 9.

Important note on file naming conventions

With time, database files will accumulate a very significant amount of information which represents many hours of work. It is therefore imperative to handle those files with great care. This includes very regular backups (copying the file(s) to a safe place, perhaps a copy to a different place on the computer and onto a flash memory stick or

[4] Where database files agree, the rater IDs present in the first file are transferred to the resolved file. The rater ID entered during the resolution process will be associated with the decisions made in the difference-resolution process. The resolved output file will contain the rater IDs of the first supplied database file where the two raters agree, and the rater ID supplied to the administrator module (plus a mark-up 'RESOLVED') where they do not.

several). It also includes keeping on top of different versions of database files. One way of doing this is by (re-)naming database files for maximum clarity (no spaces are allowed in file names to be processed, however).

To aid users in managing database files effectively, a brief summary is here given of WADP's naming conventions regarding databases:
In the categoriser module,

- newly created database files are named `db-DATE.dat`, where `DATE` is the current date in the format DD-MM-YYYY. New database files are created when a) no database file was supplied as input, b) after categorising, the user chooses to create a new database file rather than updating the old one, and c) if a .csv-formatted database file was supplied (see section 5, below).
- updated database files retain their names.
- if any filename is identical to the name of an existing file in the output directory, the new file will have `-1` or `-2`, etc. added to its name to avoid the overwriting of existing files.

In the administrator module (see below),

- if two database files are manually resolved, the new resolved database is named `resolved-db-DATE.dat`, regardless of the names of the source database files. The source database files are left as they are.
- if two databases are automatically combined, the new combined database is named `combined-db-DATE.dat`, regardless of the names of the source database files. The source database files are left as they are.
- if any filename is identical to the name of an existing file in the output directory, the new file will have `-1` or `-2`, etc. added to its name to avoid the overwriting of existing files.

5 MORE ADVANCED OPTIONS

Use of existing database files in csv-format with the categoriser module

The categoriser module of the WADP features the ability to import existing csv files with categorised word association data for use as databases. To make use of this feature, a csv file conforming to the specification for csv-databases (see appendix, page 15) can be provided in lieu of a database in WADP's own dat format. Note that the WADP will output the usual .dat suffixed database which will include items supplied via the csv-database.

Editing a database file

If categorisations in the database need to be changed for any reason, this can be achieved by editing the respective .dat file in a text editor (by performing a 'find and replace' action, for example). This should be seen as a last resort and a backup copy of the database file to be edited should first be created. This is because of the possibility of inadvertently introducing formatting (such as extra spaces, tabs or line breaks) that can cause errors in automatic processing. Changes should therefore be kept to the necessary minimum. It is essential that after any changes, the file is saved in plain text format. For a brief explanation of the .dat file format, see the appendix (page 14).

Changing the default set of categories used by the categoriser module

If changes should be made to the categories that are displayed during the categorisation process, this can be accomplished by editing the file `WADP.sh` in a plain text editor. It may be safer to make a copy of `WADP.sh` before making any changes, then open it in a text editor. The section of the file entitled 'the following section can be adjusted' and ending with the comment 'end of user-adjustable section' hold the information used by the categoriser module to display and handle category assignments. In version 1.0, these are around line 195 to 222 of the code, but this can change. Both the keys and the list of allowed categories should be changed at the same time. Only the information INSIDE of `key='...'` and `allowed_categories='...'` should be adjusted and the single quotes are essential. When editing is finished, the file must be saved in plain text format. Note, however, that the categoriser module accepts any and no category during manual categorisation (even categories not displayed as possible categories). This is to allow flexibility if needed.

6 FREQUENTLY ASKED QUESTIONS

Q1 I work with data where I have more than one response for each cue per participant. Can the WADP deal with this type of data?
Answer: Yes, it can. The key is to prepare the data in such a way that each response is on a separate line. If respondent IDs are used, each line must have a unique respondent ID, but this can be achieved by appending a response count to the end of an ID. For example if the ID is *respondent1*, their first responses to each cue might be under the ID *respondent1-1*, the second set of responses to the same cues under *respondent1-2*, etc. It is okay to leave some fields empty, for example if respondents gave different numbers of responses for different cues. An input file might then look something like this:

respondent ID	cheese	fence	rock	...
respondent1-1	moon	post	hard	...
respondent1-2	crackers	cow		...

Q2 I produced an output file without rater IDs. How can I convert it into one with rater IDs?

Answer: run the categoriser module and supply it with the *uncategorised* version of the in-file that needs to have categories and rater IDs added. Also supply the most up-to-date database file. Answer `y` when asked if rater IDs should be included. There should be no need for any manual category assignments since they should all be stored in the database, but if a categorisation was left blank at the first categorisation, the user will be asked to categorise it now. The output list will contain the desired rater IDs.

Q3 How can I use the WADP in a scenario where inter-rater reliability needs to be checked?
Answer: One suggestion for a procedure would be following:

(a) Prepare the word-association data to be categorised in a csv file according to the specifications in the appendix for in-files.

(b) Supply two raters with a) the latest version of a .dat database file and the in-file with data to be categorised.

(c) Each rater independently assigns categories to the in-file data using the categoriser module. They return the updated .dat database files.

(d) The administrator module can then be used to first produce a list of inter-rater differences and then to resolve those differences.

(e) The resulting resolved-db.dat file can then be supplied to the categoriser module, together with the uncategorised in-file to produce a categorised output file that includes the resolved categorisations. This output file can then be further processed with the reporter module.

(f) The reporter module features a menu item that allows the calculation of an inter-rater agreement ratio which shows the ratio of agreement between raters with respect to all categorisations in the input data file.

(g) The combined database file can then be used as the basis for future categorisations. If, while the two raters categorised the in-file, the .dat file was changed in some way (such as being used by a third rater to categorise more data), the two most recent versions can again be combined into a unified, up-to-date database file using administrator module.

Q4 I somehow ended up with several database files that all contain valid category assignments not contained in the others. What do I do?
Answer: Use the administrator module to combine the databases pairwise until a combined database with data from all the various database files results.

Q5 I am unsure which database file is the most up-to-date one. How do I find out?
Answer: It is sometimes possible to check the creation or modification date of files (or order them in a window according to any of those dates. If that cannot be done or database files have been modified after creation and do not provide reliable indications, it will be easiest to simply use the administrator module to combine the files that are thought to be the most recent into a single combined database file which will then contain the most up-to-date data.

Q6 Can I use filenames with spaces in them?
Answer: no, using filenames with spaces in them will most likely cause problems. Use underscores (_) instead.

Q7 I have a file with categorised responses that are not in the database. How can I add those categorisations to the database?
Answer: This is a two-step process:

(a) Make certain the file is saved as a csv file (comma separated values) and in the correct format (see the appendix on the format of the categorised output file format). Then use the administrator module and choose the (T) option.

(b) After turning the csv-file into a dat database file, the administrator module can be used again to combine the newly created database file with an existing database file, either using the (C) or (R) options.

Q8 How can I can I change the colour, font and size of the application window?
Answer: It is worth setting up the WADP's window settings so that it is convenient to work for extended periods of time. How this is done depends on the operating system and Terminal application used, but on all of them, the size of the window, the background colour as well as colour, type and size of font can be adjusted. Typically, a light background with dark font of at least size 12 or 13 pt is recommended.

Q9 I made a mistake in rating a cue - response pair. What can I do?
Answer: The rating screen (see Figure 4) has a option to go back to the previous pair to change the rating. It is only possible, however, to go back to ONE previous rating. If categorisations before that need changing, the database file needs to be edited (see page 10).

Q10 Where can I get support or answers to other questions about the WADP?
Answer: Please raise an issue at https://github.com/buerki/WADP/issues. Please understand that while I am grateful for feedback and suggestions, I may not be able to answer or address all issues.

7 LICENCE AND COPYRIGHT

Although the WADP has been extensively tested, results should always be checked for rough plausibility at the very least. The WADP is licensed under the terms of the EUPL v1.2 or later. The EUPL is an open-source license which allows licensees to use, reproduce, modify, distribute and sub-license the software. See the attached licence file or https://joinup.ec.europa.eu/software/page/eupl/licence-eupl for the full licence. Please read the licence, especially articles 7 and 8, prior to using the software. As of 2015, the copyright to the WADP is held by Cardiff University.

A appendix: format descriptions

This appendix lists file format specifications used by the WADP.

in-files (files containing word association data to be categorised)

FILE FORMAT csv (comma separated values) according to the specifications of the Network Working Group of the Internet Society, as published at datatracker.ietf.org, with the exception that newline characters are not allowed within fields.

HEADER ROW The first row of the file must be a header row. It must be formatted as in one of the following examples: (the respondent ID field is optional)

respondent ID	cheese	fence	rock	...

or

respondent ID	cue 1: cheese	cue 2: fence	cue 3: rock	...

or

respondent ID	1. cheese	2. fence	3. rock	...

If there are no respondent IDs, the first field is left out:

1. cheese	2. fence	3. rock	4. shove	...

The field labelled 'respondent ID' can in fact contain any text at all as long as the word 'ID' is included. In the cue fields, the second pattern can be used with any text preceding the colon, so 'any text at all: cheese' would be fine.

REMAINING ROWS Depending on whether a 'respondent ID' row was declared in the header, these rows take either the format

respondent 1	moon	barbed wire	hard	...
respondent 2	goat		and roll	...

or

moon	barbed wire	hard	shovel	...
goat		and roll	push	...

There may be empty fields, such as when no response was given by a respondent. Respondent IDs can be anything, but must be unique for each respondent.

dat-database files

FILE FORMAT dat (content is in plain text)

HEADER ROW none

REMAINING LINES Each line of the document consists of a cue word, followed by a tab, followed by a response in uppercase, followed by a bar (|), followed by a category, followed by a semi-colon, followed by the ID of the rater that entered this particular categorisation. This is followed by a tab and the next response, etc. All lines end in a tab, followed by a line break. Here are the beginnings of two lines of an imaginary database file:

fence	POST|M;buerkiA@cf.ac.uk	BARBED_WIRE|M;buerkiA@cf.ac.uk
irony	IRONIC|M;buerkiA@cf.ac.uk	NOT_FUNNY|M;buerkiA@cf.ac.uk

csv-database files

FILE FORMAT csv (comma separated values)

HEADER ROW The first row of the file must be a header row. It must be formatted as in the following example:

WA001	cheese	WA002	rock	...

Note that a field with 'WA' immediately followed by one or more digits, needs to alternate with fields containing cue words.

REMAINING ROWS As in the following example:

moon	OC	barbed wire	RC	...
goat	RC	and roll	CR	...

categorised output files

FILE FORMAT csv (comma separated values)

HEADER ROW The first row of the file must be a header row. It must be formatted as in the following example. The fields with respondent IDs and the fields labelled 'rated by' are *optional* and can be left out:

respondent ID	cheese	category	rated by	...

The field labelled 'respondent ID' can in fact contain any text at all as long as the word 'ID' is included.

REMAINING ROWS As in the following example:

resp 1	MOON	OC	anonymous	...
resp 2	GOAT	RC	anonymous	...

As in the header, the fields for respondent IDs and rater IDs can be left out.