

Delve: distinguishing features

This document is intended to be read in conjunction with the 'Choosing a CAQDAS Package Working Paper' which provides more general commentary of common CAQDAS functionality. It does not provide an exhaustive account of all the features provided by Delve but is designed to highlight distinguishing features and those likely to be of interest to users. The Comment section at the end details our opinions on certain aspects of functionality and usability. See also Silver & Lewins (2014) [Using Software in Qualitative Research: A Step-by-Step Guide](#). We thank Alex Limpaecher of TwentyToNine for ensuring the accuracy of this review. This document reviews Delve as of March 31st 2020.

Background <https://delvetool.com/>

Delve is an online CAQDAS tool developed by Alex Limpaecher and LaiYee Ho who are based in New York City, USA

- Delve was developed from a 'human-centered' design perspective to enable the collaborative analysis of qualitative data with tools that are simple to learn and navigate
- Delve has been available since 2018
- Delve enables the code-based analysis of text and was explicitly designed to support inductive approaches to qualitative analysis, driven by human interpretation.

Minimum System Specifications (recommended by developer)

Delve is a web-based application and is designed to work in all modern internet browsers, including Google Chrome, Safari, Firefox, Opera and Internet Explorer

- Delve is optimized for Desktop but also runs on smartphones and tablets
- Delve runs on Windows, Mac and Linux.

The Structure of work in Delve

The Delve interface comprises three main panels

- The left panel provides access to the main components - Codes, Snippets and Transcripts – the central Search feature, account information and access to different projects being worked on
- Data files are called 'transcripts' in Delve and are displayed in the middle panel. By default, the text of the transcript currently on view along with any previously applied coding is displayed. This view can be changed to display Participant Information where Descriptors can be created and applied to Transcripts (see below)
- The right panel lists the coding scheme which is always on display when a Transcript is on view. New codes are created, defined and organized in the right panel, and existing codes to be searched for by name

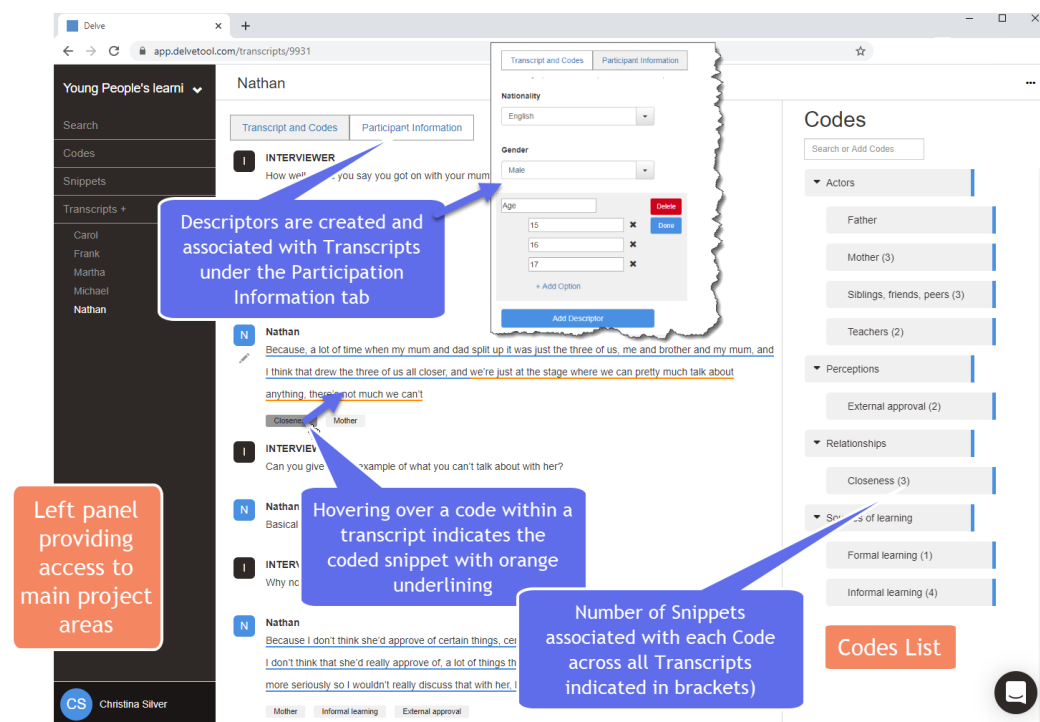


Figure 1. The Main Delve Interface

Data types and format in Delve

Delve supports textual data saved as Plain Text (.txt) ■ Transcripts can be added to a Delve project by drag and drop or copy/paste. There is no limit to the number of Transcript that can be added to a Delve project ■ Speaker identifiers contained within Transcripts are automatically identified and visualized within Transcripts ■ Data uploaded to Delve projects is stored on TwentyToNine's servers or third-party servers – see <https://delvetool.com/terms>

Closeness to data and interactivity in Delve

The Delve interface is intentionally simple and has been designed so that the different areas are straightforward to access and the features easy to use ■ Text can be edited from the middle panel, on a paragraph by paragraph level ■ Clicking on either the Search, Codes and Snippets areas (from the left panel) changes the middle panel to display relevant features enabling detailed focus on each set of features ■ Clicking on a code within a Transcript brings up a floating window on top of the main interface displaying all snippets across all transcripts so far associated with that code, along with the code comment field for adding definitions and interpretive commentary. The same information can also be accessed at individual codes from the Code List (see below) ■ Browser back and forward buttons take the user to previous views within Delve, not out of the application ■ Interactivity between the different areas and views is good

Coding scheme in Delve

Codes are created in the Code List (right panel) when viewing a Transcript ■ The coding schema can be as hierarchical or as un-hierarchical as required ■ As many codes as required can be created, a hierarchical structure can contain up to three levels, and data can be coded at any level in the hierarchy ■ Sub-codes can be hidden and revealed within the Codes List ■ Drag and drop allows easy re-organisation of codes across and within hierarchies ■ Codes can be renamed and deleted as required ■ Code definitions can be associated with Codes in the Details pane – accessible both from the Code Page (left panel) and the Code List (right panel) ■ Code definitions and the number of Snippets applied to each Code across all Transcripts are displayed in when viewing the Code Page

Coding Processes in Delve

Sentences can be auto-selected with one mouse click and paragraphs using a short-cut icon adjacent to the text ■ Clicking on a code when text is selected applies that code to the selected text ■ Segments of text that have been coded are called 'Snippets' ■ As many codes as required can be applied to the same or overlapping Snippets

Basic Retrieval of coded data in Delve

Snippets that have been coded are underlined in blue within Transcripts providing a visual overview of all text that has and has not been coded ■ Codes applied to Snippets are displayed as icons below each paragraph of text within Transcripts ■ Hovering the cursor over a Code icon within a Transcript changes the underlining to orange for the Snippet to which that Code is applied ■ Selecting a coded paragraph within a Transcript highlights all the codes applied to Snippets within that paragraph in the Code List in the right panel ■ Clicking on a code icon within a Transcript retrieves all the Snippets coded to that Code across all Transcripts in a floating window. ■ Any other Codes applied to the Snippets retrieved in this way are also shown in the floating window, and it is possible to navigate directly from that display to see the coded Snippets within their source Transcript context

Data Organisation in Delve

Descriptive organization of data according to known characteristics (for example socio-demographic characteristics of participants) is enabled by applying Descriptors to Transcripts in the Participant Information tab ■ Descriptors and their values can be created and applied to Transcripts at any time ■ Snippets can be filtered by combinations of Descriptor values together with combinations of Codes and Transcripts via the filtering tool (see below)

Writing Tools in Delve

There are two writing 'spaces' in Delve – Transcripts and Code descriptions ■ Empty Transcripts can be created within a Delve project to serve as memos, and the text within them can be analysed using the same tools available

for other Transcripts ■ Code descriptions are associated with Codes and are viewable and accessible from wherever the Codes appear. There is no limit to the character length of Code descriptions



Figure 2. Retrieval and Filtering in Delve

Searching and interrogating the dataset in Delve

Searching in Delve refers to finding a user-specified word or phrase within Transcripts that have been added to the project. Paragraphs in which the searched-for word or phrase occurs are listed, with the search term emboldened ■ Interrogation happens in Delve by filtering Snippets according to combinations of Codes, Transcripts and Descriptor values. Opening the Snippets page initially displays all Snippets created across all Transcripts, listed in ascending order according to when they were created. Snippets can be filtered according to the Codes which are applied to them, and any number of Codes can be filtered to. For the Snippets of multiple codes to be retrieved via filtering in this way, all the selected Codes need to be applied to exactly the same sized Snippet.

Output from Delve

Transcripts, Codes and Snippets can be exported from Delve ■ Transcripts can be exported in Plain Text (.txt) format or selected text from within Transcripts can be copied and pasted into an external file ■ The list of Codes can be exported as a MS Word File (.doc) or as a Comma Separated Values (.csv) file. Both these exports provide information about the hierarchical position of each code within the schema (order in the List and nested level), the number of Snippets across all Transcripts associated with each Code, and the Code Description ■ Snippets can also be exported into a .csv file, filtered as required to codes, transcripts or descriptors ■ Embedded within output are unique urls for snippets which enable quick access back to the Delve project

Teamworking in Delve

Delve is designed around the need for multiple researchers to work concurrently on a research project. Collaborators can be added to projects at any point and access work by logging into a Delve project they have been given access to, using a password.

Comment on Delve

Delve has a simple, clean and intuitive interface and is easy and quick to learn and use – familiarizing with Delve took us just one evening! It is therefore a good choice for those new to undertaking qualitative analysis using dedicated digital tools, who need to get going with work quickly. Our comments are therefore made in the context of the simplicity the developers seek to provide.

Delve is a good option for teaching qualitative methods and technologies together and the monthly price structure with [educational sector discounts](#) for students, teachers and academics will be attractive to many users.

It is straightforward to import data into Delve using drag/drop or copy/paste and textual data is displayed in a convenient and visually appealing way. The ability to easily auto-select sentences and paragraphs within Transcripts is helpful in speeding up processes. The current ability to only work with plain text may be restrictive for some users, but Delve have plans to include other forms of qualitative data in the future. Some users may also be concerned about where data are uploaded to, so are advised to review information on data storage and privacy (<https://delvetool.com/terms>).

Transcripts are listed conveniently within Delve and always on view within the left panel, facilitating seamless access. This works very well for small datasets, but once working with more than a dozen or so transcripts becomes a little cumbersome as transcripts cannot be re-ordered or otherwise grouped in this panel.

The ability to create empty transcripts and edit them at any time is useful for several purposes. For example, transcripts can act as research journals or analytic memo that can be analysed using the same tools as used for transcripts containing primary or secondary data. This allows for good integration between data and interpretation and is a welcome feature. However, the lack of a special note-taking panel means the user is required to think carefully about how to most effectively manage interpretations through writing.

Delve includes standard tools for code-based analysis of textual data which are implemented neatly and intuitively. The coding of Snippets via just one click will be particularly valued by users. However, un-coding snippets, although possible, is a little cumbersome to achieve in comparison to other code-based tasks.

Searching for words and phrases using the Search panel is straightforward and there is good interactivity between the list of results and the source context. However, the current inability to automatically code hits to a user-specified unit of context (e.g. a sentence or paragraph) is restrictive for approaches that focus on the occurrence of keywords.

The retrieval of coded snippets is visually appealing and easy to navigate. The display of codes applied to snippets immediately underneath them within transcripts is particularly helpful, although it would also be useful to be able to temporarily remove this display as required. The change in colour of underlinings when hovering the cursor over a code displayed within a transcript is also very helpful.

Delve has good interactivity between panels and aspects of analytic work. This is particularly evident when filtering snippets based on the application of codes in relation to descriptors. The filtering principle is also very straightforward to comprehend and use. However, there are currently some restrictions in retrieving snippets based on the combinations of codes because of the need for codes to be applied to exactly the same snippet in order to be retrieved via filtering. It would be a pity to overload this straightforward package with complex query features, however a simple overlapping codes (a loose Boolean “and”) or all selected codes (a Boolean “or”) combinations would be simple and useful variations in the retrieval of coded data.

Being able to add collaborators to a Delve project so that multiple researchers can work on the same analysis project at the same time greatly facilitates team-work in easy to use ways. The Delve team have several additional collaborative features planned. Two improvements would be the ability to see who logged on last, and an integrated between-researcher chat-type feature. Knowing ‘who did what’ within a project and possibilities to

compare the coding of different users would add transparency. Snippet urls within output is a unique feature and supports the sharing of analysed qualitative data and collaboration.

*The **support materials** available from the Delve website are easy to access and helpful when getting started. The ability to start a conversation with the developers – to for example ask a question or provide feedback – from within a Delve project will also be helpful for many users.*

Further Reading

Delve is relatively new so at the time of writing we are not aware of any peer-reviewed publications about it. However, the [Delve blog](#) has several useful posts which will supplement the information provided within this review and the following is a case-study on how user-experience researchers have used Delve:

Liam Cates [How DonorsChoose Uses Delve to Uncover Opportunities to Help Teachers](#), November 25th 2019