

# Conlang Database Project

## Modus Operandi

Matthew McVeagh, completed 2020/01/06

### I. Basic setup

There would be a front-end website and a database behind it. The database would take the form of a simple two-dimensional array, with a set of conlang entries which each have data in at least 13 fields, with possibly more if needed for technical reasons. The database could be added to by the public ('contributors') by data entry from the website, and also by more direct editing by project members ('compilers').

The database data would be regularly backed up in a couple of places to prevent loss. In the event of loss or extreme damage to the database these backups would need to replace/restore the database, so their capacity to do that should be checked from the outset.

Conlangs submitted by public contributors would not immediately be part of the approved database contents and included in search results. Instead they would first be checked by project members, to be known in that role as 'reviewers'.

The process of adding conlangs en masse by compilers would hopefully be more convenient than submission by the public input form, and this might make some examples of the job easier, for instance if a lot of languages entered in a session shared common data (same creator, year, type etc.). Something like the project spreadsheet might be useful, especially if it displayed some of the database already existing for comparison.

In both the cases of public contribution and admin compilation there should be a flag if it seems a language being entered is already on the database, e.g. if it has the same name as one on it. But sometimes different languages have the same name so there should be a capacity to proceed anyway, and the database should add a "[1]" and "[2]" etc. to the end of such names in some order to be decided.

The project is intended to benefit the conlang community, and is not for profit; work done for it is voluntary and its products would be licensed under "Creative Commons". It would be perfectly legitimate for someone to copy the whole database contents, or any subset thereof, and for instance to fork into a new project, perhaps with slightly different parameters. Although at least to begin with it would be the only project of its kind it would not presume any kind of 'official' status.

### II. Website features

The website need not be too big or ambitious. It would include:

1. an introduction and explanation of what the project and website are about
2. a search feature for members of the public to query the database and get a report of results, with guidance on how to use it including field and text types, logic of combination etc.
3. an input form for members of the public to enter conlang data to be added to the database, with guidance on how to do that, including definition of terms, prioritisation of fields, etc.

4. a contact form for members of the public to send corrections, revisions/updates, comments, suggestions, questions, complaints etc.
5. optional extras such as articles on the database contents, links to other conlang pages, etc.

These could all be located on the one main page if possible, or if not the search and input functions could be linked to on separate pages.

In the longer term we might consider enabling non-English versions of the website.

Taking each of those features in turn:

## **II.1. Introduction**

This should be brief; the main purpose is to get a reader who has no idea what the website is about to understand quickly, or someone who does know the site to recognise that it's not some other site they clicked onto.

## **II.2. Search feature**

### **II.2.a. Search feature appearance and mechanics**

This is the main function of the site and it should be prominent. Ideally it or the link to its page should be higher up the main page than the input form.

Users should be able to search any of the database fields in any combination. Therefore input boxes for all fields should be visible, in their order in the database. Entering search terms in a box would mean that the contents would be included in the search filtering, leaving it empty that it is not. There should also be a box for general keyword search, which would look for the entered string in all fields.

It's possible, in fact likely, that many users will want to make simple searches of only one or two frequently repeated fields, in which case it may be worth developing a scaled-down 'simple search' as well as the all-inclusive one. The more commonly used one can then be made default, with the other one linked to on another page or opened up by clicking a button.

The fields contain different types of data text (e.g. Start Date is year numbers, CRC code is a short sequence of letters, Notes may contain extensive lines, Name of Language in Itself might be in a conscript, four of the fields will have limited menu options). It would not be absolutely necessary for the wrong type of text not to be allowed in a typing box (e.g. if only typed numbers appeared in the Start Date box, and typing letters or punctuation resulted in nothing) but there perhaps ought to be some indication that some box entries wouldn't lead to valid results. At least the guidance should cover this.

With regard to fields of variable text content, it should ideally be possible to enable wildcard searches. E.g. if a searcher knows a conlang they're looking for has a name beginning with "Ast..." or that it's from the 1990s, they should be able to enter "Ast\*" or "199\*" and get relevant results.

The limited option fields should be done by checkbox-style choice from a list of options rather than a typing box. Some such fields are single-choice for an individual conlang's database record, although Type(s) is now going to allow for multiple choices. However that limitation shouldn't apply to the search feature, as users should be able to search for several options at once per field. There could be a 'select all' checkbox at the top of each list, checking which would choose all options,

unchecking removing all. Users could then more easily search for "all but one" or "all but a few" options.

### **II.2.b. Search feature guidance**

Searchers will need a fair amount of guidance on how to enter search terms to get what they want. There should be a brief text before the form boxes giving general hints, such as that fields can be combined, that the keyword search allows wildcards and covers all fields, that only certain kinds of entered characters will be valid in certain fields (numbers in Start Date etc.).

Besides this I think it would be a good idea if there were also tooltip-type pop-up boxes giving more specific help on each field box. Perhaps rather than strictly tooltip they could be activated by clicking a '?' icon. These would explain briefly what the field is for and make clear any particular circumstances for search terms (characters allowed, length, wildcards etc.). In the longer term we should probably write separate pages explaining each field and the type of content in it, which could be linked from the pop-ups.

### **II.2.c. Search results display**

It's a good question how the search results should be displayed. It seems unlikely to me that it would be practical to try to display them on the same main page; instead a new page would be generated with the results listed on. If the results extend beyond a certain number of lines there should be a cut-off and links to subsequent pages giving more of the results. Any results page should have links to the main page and any other relevant ones; a list of the search terms used; and another search form at the bottom for the user to try a different search. Possibly the previously used terms could be auto-suggested in the bottom search form.

It's also a good question how the results should be displayed in terms of what data should be included. E.g. should all fields' content be included, even those that haven't been filtered for? Considering there will be 13 fields, this would undoubtedly clutter up the page horizontally. Perhaps by default only data from the fields filtered for, plus some identifying fields such as names of conlang and creator, would be displayed. If the searcher wants to include unfiltered fields in the display perhaps that could be checked in the search form.

In the case of fields having more than one line, e.g. conlangs or creators with multiple names, they should be stacked vertically.

The ordering of entries within search results by default would be by conlang name. But again it might be possible to allow the searcher to order by a different field, such as Date or Name of Creator. If this could be done within the generated results, so a searcher could experiment with different orders to see which was best, that would be better than trying to fix it from the initial form entry.

Each conlang entry in the results should have a link that clicks through to a page that displays all the information about that one language, preferably vertically. It should be possible to go back from that page to the same search results without getting an error message.

Any search result page should be conveniently downloadable or printable.

### **II.2.d Other search matters**

Another feature we've previously imagined as part of the database service is the production of statistics relative to particular queries. If we do implement this we'd have to think how to work it in relation to the other search function mechanisms. For instance do we have an option on any search form to give only the number of hits instead of listing all the conlangs? Or to give the number as well as listing them? Should the number of hits from a search be an automatic part of the search results display - e.g. "199\*" -> "43 conlangs" and then the list of them. What about if

someone wants to know relative/proportional statistics, e.g. how many conlangs overall are personal languages, jokelangs, auxlangs etc. This would require a different sort of query and results display entirely, which we would have to think about separately.

### **II.3. Input form**

This is the second-most important function of the site and so it or the link to its page should be next below the search form.

The input form will enable members of the public to contribute conlang data to the database, without having to be project members and get a login to edit the database directly. It should be attractive enough and not too off-putting in scale or complexity, for which reason we will have to think about its presentation and guidance given.

As with the search form there will have to be a box for each field. In this case where a limited option field allows only one option to be chosen, the option list should be more like a radio button than a checkbox type. Again only certain character string types should be allowed in certain boxes; in cases of possible multiple entry such as the names of conlangs and creators there could be a feature where another line of typing space appears if the contributor presses Enter at the end of a string. The Notes field box should be particularly large, indicating the contributor can type more in that space (perhaps an upper limit of 5000 characters).

Once again there should be a general guidance text preceding the form, and tooltip-type pop-up info boxes to help contributors with specific fields. The general text should emphasise that it's better to get some info sent than none, such that if there is something they don't know or are not sure about it's fine to leave a box empty. The most important data being the conlang name and a link which project members can use to find the information themselves. It should also be mentioned that mistakes and later changes to the conlang can be corrected by means of the contact form, and that the Notes field is good for details of what makes the language distinctive and further information not included elsewhere e.g. what exactly the language is when it's marked as 'Other' in the limited option fields.

Upon submission and receipt of one conlang's data, the page should refresh with a message describing the result (successful admission, or some problem) and a new input form for if the contributor wants to submit another. Again if this new form could auto-suggest the data from the last submission that would help contributors who would like to enter several websites with data in common (e.g. same creator).

There are two ways the form-inputted data could be received by the website:

(1) into a second (small) database, where the details could be checked, corrected, added to etc. and then transferred to the main database.

Or (2) directly into the main database, but with a flag such as 'Pending', which would mean those entries wouldn't be included in search results and instead would be subject to the same checking. After this the 'Pending' flag would be removed and they would be included in search results.

### **II.4. Contact form**

The contact form should be fairly simple and even if the search and input forms need to be moved to separate pages there is less reason for this with the contact form.

There should be a main large box for the main message, plus a couple of small ones for the contactor's name and email address, and maybe a drop-down menu for the message type (question, complaint, suggestion, conlang update/correction, fault notification, etc.). However

people should not be forced into applying message types they don't believe are right so there should either be an "Other" option or the whole message type choice should be optional.

The form should direct to a specific admin email address, something like `contact@database.conlang.org`. That can be accessed by one or more relevant organiser.

The form should be preceded by a brief guidance text suggesting different possible reasons for contacting, explaining the consequences of including or omitting things like an email address, and giving a response time estimate.

## III. Participant roles and activities

### III.1. Roles

There would be at least four roles for those involved in the project:

**Contributors** as already noted will be members of the public submitting conlangs via the public input form on the website. In order to avoid putting off contributors we could not insist on people entering such data having an account, so contributions would effectively be anonymous (although we could include a box in the input form for the contributor to enter an email address in case of need of further contact). Contributions could however be identified by timestamp and IP address.

**Reviewers** will look over public contributions and check them for appropriate content.

**Compilers** will actively go out to search for as-yet un-added conlangs and add them direct to the database.

**Organisers** would have an executive role, setting policy and direction and co-ordinating the others.

Contributors wouldn't count as 'project members'; however reviewers, compilers and organisers all would.

The same person could have any combination of these roles, e.g. someone could do both reviewing and compiling work, could be an organiser overseeing database management and also do some contributing or compiling, etc.

### III.2. Activities

Issues for reviewers to check for would include missing data, incorrect data, anything ethically or legally objectionable, and consistency with the standards and settings of the whole project. Checking might involve searching for independent information on the language, contacting the language creator, and in difficult cases might be referred to the organisers and wider group of reviewers.

Compilers would go about adding conlangs methodically, preferably without duplication and with some prioritisation. They would each arrange to focus on a particular source with the organisers. This would hopefully avoid overlap, while being acceptable to each compiler (some people might like to focus on a source they're familiar with, e.g. a particular wiki, forum, linklist etc.) and ideally proceed over time from the simplest and most obvious sources to ones more difficult to glean from. As the job of compilation would be a more in-depth submission of data than public contributions via the input form there ought to be a physical way of inputting that is more efficient or convenient (they might be doing 20 or 30 in a sitting, and it might be good to do some simultaneously if for instance several have something in common).

Organisers would check the quality and quantity of what's being added, and direct the active compilation and its prioritisation. They would manage the project team and any technical challenges (thus they would need to include people specialising in database management, website coding etc.), including possible improvements to the website and database. Organisers would also have the roles of responding to public contact, e.g. suggestions, queries, complaints, requests for correction etc., and represent the project to the world, including publicity and liaison.

There would need to be good communication between all project members, in different subsets of the whole. To this end it would be good if there was a message/notification service that let members of the team contact each other to discuss the workflow. Although it should also be kept functional as all project members will be busy with other things so will not want too much irrelevant messaging. Ideally the different roles would each have their own appropriate level and scope of powers, such that people don't have power to alter parts of the database or website that their role is not concerned with, for instance. In the case of crucial roles there should be understudying so another person can step in if the usual one can't do the job temporarily or permanently. Decisions made should come with a record of who made them, at all levels.

I would suggest a weekly review of progress, which could lead to changes such as aiming to add more team members, shifting compilation focus, improving policy in the light of feedback etc. It would be a good idea for both reviewers and compilers and even some organisers to decide on a weekly commitment level, and then adjust that according to circumstances.

### **III.3. Longer term**

Although I (Matthew) am happy to head up the project now I know I will want to step aside eventually and let someone else take over, so there needs to be some transferability and legacy consideration. It would be best if executive roles were shared and there was not too much dependence on any one individual.

## **IV. Setup sequence**

We'll have to give some thought to how exactly we first set the website and database up.

First we will have to build a basic website, and add people as admins with some kind of login to do work on the website and database. Whatever we build will have to be tested before being publicly released. A lot of what I've described in this document is 'bells and whistles' rather than the most basic, important factors, so those sorts of things can be deferred till later.

We will need to import the existing public spreadsheet to the database, en masse. We should also stop public access to the spreadsheet as soon as we've done so, because otherwise new conlangs will be added to that and it would be confusing as to which are newly added and which were already there. But when we close public access to the spreadsheet we should have the website input form working, as that is what contributors will have to switch to. Therefore I propose we create and test the input form as a means of adding new conlangs first, *before* we copy the spreadsheet contents and close it to the public. An alternative is to empty the spreadsheet after copying (having carefully backed up by downloading, of course). I think this would not look as encouraging for contributors coming along to add languages.

We should probably get up to speed reviewing and approving all conlangs submitted so far before we embark on searching and compiling from wider sources. However if we find we're constantly running to catch up we will need to find a way to add more volunteer hours to the project so the compilation process can start.

END OF DOCUMENT