Biodiversity Volunteer Portal

# Preparing BVP export data for import into EMu

## Process

This document details the manual and semi-automatic processes of taking transcribed label data from the Biodiversity Volunteer Portal and putting it into a form for importing into EMu, the Australian Museum database.

John Tann
Australian Museum

December 2013

# Contents

# Chapter 1 Preliminary

Some things need to be done ahead of others. The order is important.

Do as much manual work in Excel as possible before passing it through automated procedures. You don't want to re-run an automated procedure only to find you need to follow it with a repeat of a manual operation.

## Prepare BVP spreadsheet

BVP data is downloaded as a series of CSV files. For example, 'BVP treehoppers original.csv'

In Excel, open 'BVP treehoppers original.csv'. Save as Excel spreadsheet 'BVP treehoppers repaired xx.xls'

Change spreadsheet column headings in 'BVP treehoppers repaired xx'. Add some columns.

| **BVP heading** | **change to** | **comment** |
|---|---|---|
| | sequence | create – column 1 |
| ocean | LocOcean | create for Malacology (not Ento) |
| country | LocCountry | rename |
| stateProvince | LocProvinceStateTerritory | rename |
| | LocDistrictCountyShire | create – for islands |
| | LocTownship | create |
| verbatimLocality | verbatimLocality-original | copy, rename and move to penultimate |
| verbatimLocality | LocPreciseLocation | rename |
| | | |
| samplingProtocol | samplingProtocol-original | copy, rename and move to end |
| samplingProtocol | ColCollectionMethod | rename |
| | ColEventCode | create for event codes |
| scientificNameAuthorship | originalNameAuthorship | rename – scientific name author not recorded |

Do not touch **locality**. This is a Google attempt at the placename, and will confuse.
Use spreadsheet tools to sort the data and:

- Everywhere. Global replace '\n' with a space. Remove double spaces. Remove Â.
- Add **country, state**, where indicated
- Column **LocPreciseLocation**: Correct obvious spellings and omissions in. Remove in-full state names. State abbreviations will be removed by scripts. Add a comma where helpful, to aid parsing – eg Manly, Sydney. Change Sth Coogee to South Coogee.
- For PNG, USA and others, add states, districts (check against SITES picklist if possible). Move countries out of column **LocPreciseLocation.** See below for modern names of countries, and provinces of PNG.
- Column **LocPreciseLocation**: Put "Site 4" and other Collection Event Codes such as "BRITTON 2008/001" into new column: **ColEventCode**
- Column **fieldNumber**: Move "Site 4" and other Collection Event Codes such as "BRITTON 2008/001" into column: **ColEventCode**. Delete from column=**fieldNumber**

- Column **LocPreciseLocation**: Put Lord Howe Island, other 'stand-alone' islands , and island groups under the new column: **LocDistrictCountyShire** (check against SITES picklist)
- Column **LocPreciseLocation**: Remove elevation and put into **verbatimElevation**
- Put Antarctica in **LocProvinceStateTerritory** as 'Australian Antarctic Territory'
- Column: **verbatimLatitude** and **verbatimLongitude** – replace "Â"
- Column **verbatimElevation:** Remove commas, dots

Save as Excel spreadsheet '**BVP treehoppers repaired xx.xls**'. This spreadsheet is added to by DATES procedure and other procedures will subsequently modify that.

Further manual modifications should be made to '**BVP treehoppers repaired xx.xls**' and saved.

Follow these changes with the DATES, METHODS, PARTIES, SITES and EVENTS procedures. Note that the outcome of DATES procedure will not need to be cut and pasted a second time.

## Other tasks

**Collections**
In column: **occurrenceRemarks** filter='coll'
Go through the collections and add them one at a time
In column: **occurrenceRemarks** filter='Ashton'
In column: **collectionCode**
"Ashton Collection"
And so on for other collections:
"H. Ashton Collection"
"Fruhstorfer Collection"
"M.S. Moulds Collection" (Care, don't mistake 'Coll. M.S. Moulds' for a collection)
"G.A. Holloway Collection"
"Hangay Collection"
"E.H. Zeck Collection"
"W.W. Froggatt Collection"
"D.A. Doolan Collection"
"Lord Carmichael Collection"

**Collector/collections**
In column: **collectionCode** filter = 'collection'
In column: **recordedBy** check that the collector hasn't been mixed up with the collection and fix.

**Event codes**
In column: **occurrenceRemarks**
Search for things like [98-28], BRITTON 20070711, Site, PILB081/LT7
Put square bracket around the event code.
Put into column: **ColEventCode**
In BVP2 column: fieldNumber copy relevant CECs into column: **ColEventCode**
Leave numbers like 59, #259, HH79, but copy stuff like Site A, BRITTON 20070711

**Methods**

In column: **occurrenceRemarks**
Search for things like light, night, net, hand, trap, pan, mv, bred (now method=Reared), pupa, larva
Adjust in column: **colCollectionMethod**
This process will be followed up with Chapter 3 Methods, but the aim here is to get rid of anything that is not an obvious method, and include things that are.

**Types**
In column: **occurrenceRemarks**
Search for 'type'.
Adjust in column: **typeStatus**

**Elevation**
In column: **occurrenceRemarks**
Search for things like 3000', 3000 ft, 'ft', 450m
Adjust in column: **verbatimElevation**
Elevation range (eg 2000-3000 ft): Create another column: **verbatimElevationTo**
Then put lower elevation in **verbatimElevation** and higher elevation in **verbatimElevationTo**
Remove Approx. Example 'Approx. 450 m' becomes '450 m'.

**Depth**
Copy column: maximumDepthInMeters to maximumDepthInMeters-original
Copy column: minimumDepthInMeters to minimumDepthInMeters-original
In columns: maximumDepthInMeters and minimumDepthInMeters
Convert fathoms to metres.
Leave units as " m".

**Latitude and Longitude**
In column: **verbatimLatitute** and **verbatimLongitude**
get rid of spaces
replace(value, " ", "")
replace(value, ":", "°")
clean up missing degree symbols, and others

Check for completeness
Have we got all the verbatim lat-longs?
Sort on **verbatim lat-long**
In column: **occurrenceRemarks**
Search for: ° (degree symbol) below the last lat-long
Are all locations geocoded?
Sort on **verbatim lat-long**
Sort on **decimal lat-long**
Sort on **eventID**
Sort on **locationID**
Any locations left are missing lat-long

**recordedBy**
First, import names of collectors – see next section below
In column: **recordedBy**

Fill in those people that are missing:
eg Filter on empty **recordedBy** and scan through column **occurrenceRemarks**

**identifiedBy**
In column: **identifiedBy**
Fill in those people that are missing:
eg in column **occurrenceRemarks**
Search for terms like ID, DET, 'by',
Confirm date and originalNameUsage is filled in where possible for identifiedBy

**Location**
In column: **LocPreciseLocation**
Find the missing places. Filter on Countries, States comparing empty ones
Look for potential trouble with m = miles, or m=metres
Malacology alert: 'xx m' often means a depth.

**Dates**
Dates are often poorly interpreted. Roman numerals will get mistaken for something else.
See Chapter 2 Dates for some manual fixing ideas using OpenRefine
Look for spaces and remove. These are not removed automatically.

**taskID**
Run 'JSON V-WEB weblink to BVP task'
This procedure will create a link to the BVP task web page of each record. This is an aid to manually checking individual records throughout the many procedures which follow.
(I have stuck this procedure in with DATES. It is a bit out of place, but it means that it won't need to be run at a different time.)

**Catalogue numbers**
Check for duplicates
Matthew has written a function (reg2irn) to convert registration numbers to a corresponding cat irn by looking up EMu from within a spreadsheet.
This can be done in two ways

1) Cut and paste
   Use dedicated spreadsheet with **reg2irn** embedded, IMu_cat_irn_lookup.xlsm
   Cut and paste a column of catalogue numbers into column A of IMu_cat_irn_lookup.xlsm
   Copy the first data cell in column B all the way down (double-click on the bottom RH corner)
   For 1000 records this may take some time.
   Copy and paste-values into working spreadsheet.

2) As a function
   Within the working spreadsheet add in the function **reg2irn**

   Import function
   Developer | Visual Basic
   File | Import | … navigate to "C:\Users\john.tann\SkyDrive\Documents\BVP data refining\Excel macros - use alt-F11 + alt-F8"
   Class1.cls
   clsRunApp.cls

CmdOutput.bas
MHIMu.bas

Operation
Create a column called **cat_irn** next to **catalogNumber**
In first cell add the function =reg2irn(A2)
Copy the first data cell in this column all the way down (double-click on the bottom RH corner)
For 1000 records this may take some time.
Create another column to the right of **cat_irn**. Copy from cat_irn and paste-values into new column. Delete **cat_irn**. Rename new column cat_irn.

Headings of spreadsheet

| **catalogNumber** | **cat_irn** |
|---|---|
| K.338081 | |

**Scientific name**
No longer used. EMu is not interested in previous identifications

Use OpenRefine to check spelling
In column: **scientificName** facet / Text facet

1. Facet by name
2. Select xx choices
3. cut and paste the names into a spreadsheet and then into ALA sandbox

**Sex**
Controlled vocabulary is 'male', 'female', 'unknown'
In column: **occurrenceRemarks** change whacky symbols into either 'male' or 'female'
™, = male, ™€ = female
Change ♂ to 'male', ♀ to 'female'

**All other columns**
Correct spellings
Look to homogenise where possible.

## Import names of collectors

Done within a spreadsheet.
This doesn't work in OpenRefine not handling lookup lists well.

BVP2 returns a separate list of collectors called recordedBy.csv
This CSV file needs to be re-formed.

recordedBy.csv is a list of ALL the collectors.
In Excel:

1. Add a column **sequence**
2. re-parse column=**recordedBy** into four separate columns: collector1, collector2, collector3 and collector4.
3. Add the following 6 headings to the last four columns. Then paste the relevant formula into the cell below the heading.

Formula

| column | formula | comment |
|---|---|---|
| collector1 | =IF((D2=0), E2, "xxx") | copy all the way down |
| collector2 | =IF((D3=1), E3, "xxx") | copy all the way down |
| collector3 | =IF((D4=2), E4, "xxx") | copy all the way down |
| collector4 | =IF((D5=3), E5, "xxx") | copy all the way down |
| sumCollectors-formulae | =CONCATENATE(F2, " \| ", G2, " \| ", H2, " \| ", I2) | copy all the way down |
| sumCollectors | values only of sumCollectors-formulae | cut and Paste-Special replace " \| xxx" |

4. Copy the whole worksheet and Paste-Special only the values into a new worksheet. Rename the worksheet 'values without formulae'. Move that worksheet to the beginning.
5. Save spreadsheet as "BVP bee flies recordedBy list xx"
6. In OpenRefine
   Import recordedBy reference: "BVP bee flies recordedBy list xx"
   In file: "BVP bee flies cleaning xx":
   a. Run N-RBL lookup recordedBy

This adds a column **recordedBy**. Contains names like: G. Daniels and A. Daniels
The vertical bar '|' should have been replaced by 'and'.

Slow way: Use Excel. Import values from column **sumCollectors** into "VP bee flies cleaned xx". The column lengths don't match, so do it manually.

## Countries

Modern names. These are mostly repaired in SITES module JSON PB.

| Old country | New Country | Comment |
|---|---|---|
| British New Guinea | Papua New Guinea | |
| New Guinea | Papua New Guinea | |
| German New Guinea | Papua New Guinea | |
| TPNG | Papua New Guinea | |
| Territory of Papua New Guinea | Papua New Guinea | |
| Papua (SE New Guinea) | Papua New Guinea | |
| Deutsch New Guinea | Papua New Guinea | |
| | | |
| Dutch New Guinea | Indonesia | Province=Papua |
| Irian Jaya | Indonesia | Province=Papua |
| West New Guinea | Indonesia | Province=Papua |
| West Papua | Indonesia | Province=Papua or West Papua |
| | | Change country to Indonesia. Leave out 'Irian Jaya', leave out Dutch New Guinea |
| | | |
| New Hebrides | Vanuatu | |
| | | |
| Republic of South Africa | South Africa | |
| British East Africa | Kenya | |

| Old country | New Country | Comment |
|---|---|---|
| | | |
| German East Africa | Tanzania | with Burundi and Rwanda |
| Deutsch-Ostafrika | Tanzania | with Burundi and Rwanda |
| D. O. A. | Tanzania | with Burundi and Rwanda |
| | | Change country to Tanzania. Leave out 'German East Africa' |

## Papua New Guinea

PNG is partitioned into Provinces, Districts and Subdistricts. The words 'Province' and 'District' have been used interchangeably.
Leave out the words 'Province' and 'District' (except for National Capital District)

| Province of PNG | | |
|---|---|---|
| Central | | |
| Simbu (Chimbu) | | |
| Eastern Highlands | | |
| East New Britain | | |
| East Sepik | | |
| Enga | | |
| Gulf | | |
| Madang | | |
| Manus | | |
| Milne Bay | | |
| Morobe | | |
| New Ireland | | |
| Oro (Northern) | | |
| Bougainville | | |
| Southern Highlands | | |
| Western (Fly) | | |
| Western Highlands | | |
| West New Britain | | |
| Sandaun (West Sepik) | | |
| National Capital District | | |
| Hela | | |
| Jiwaka | | |

Keep the brackets.

# Chapter 2 Dates

Dates are needed to define an event.
Generally they are in pretty good shape, but come from the BVP in a special format and need to be deconstructed.

Excel treats dates very poorly, and some manipulation is needed before using them in a spreadsheet.

## Actions

1. Some cleaning of dates in the EVENTS picklist helps. A separate reference picklist for DATES is not really required.
2. For data exported from BVP, parse the date field into begin and end date.

## Files used

| Filename | Dependency | Comment |
|---|---|---|
| BVP treehoppers original | From BVP | Input file |
| Event dates changelist xx | Changelist for EMu events | Reference |
| BVP treehoppers repaired - dates xx | Action file for PARTIES procedures | Output file |
| EVENTS-ento reference – dates xx | JSON V-DL lookup matching event dates | Used for seeing how many events dates match. See EVENTS |
| JSON V-WEB weblink to BVP task | taskID | creates a link to BVP task page |

## Dates cleaning in EVENTS picklist

For EMu EVENTS
Use Excel to create a changelist of dates: “Event dates changelist xx”, with one column ‘bad and another column ‘good’.
Create a OpenRefine project called: “Event dates changelist xx”, where xx is the latest version. Note: do not repeat version names.

See ‘Regular expressions for events’ for more details

## Clean BVP dates and split into begin and end dates

### Clean BVP dates data using OpenRefine

Excel mangles dates. This is a fix.
Import CSV file directly into OpenRefine – note that dates are not screwed

**Input file**: ‘BVP treehoppers original.csv’. Create a project in OpenRefine
The order will be important, don’t re-order.

**Procedure**: z-stack-check-and-split-dates.json

This set of procedures does the following:

1. Apply the following JSON scripts:
    a. D cleaning dates

**Export** as Excel file. Don't rename yet.

Use Excel

1. **Manually** strip out 3 columns:
    b. eventDate
    c. eventDateBegin
    d. eventDateEnd
2. … and insert those columns into the working BVP spreadsheet: "BVP treehoppers repaired xx". Replace the old column: 'eventDate' with the stripped out one.

Hyperlinks

3. Manually strip out one column: **occurrenceID** (about column B) from the Excel file exported by OpenRefine
4. … and insert in column 2, before **taskID** (into the working BVP spreadsheet)
5. Move column **taskID** to the end. (See below for active hyperlinks)

6. Save Excel file as 'BVP treehoppers baseline xx'
7. In Excel, clean up any aberrant dates.

"**BVP treehoppers repaired - dates xx**" is the BVP data file with extra columns due to dates.

Follow up. If further manual changes need to be made to the original spreadsheet '**BVP treehoppers repaired – xx'**, then running the above dates procedures on that spreadsheet should be OK and you do not need to do the last 3 steps above. ie no need to strip out columns in the spreadsheet.

## General fixing

Dates are often poorly interpreted. Roman numerals will get mistaken for something else.

**Slow and manual procedure**

In column **occurrenceRemarks**, use OpenRefine:

1. Roman numerals: search for \bi\b, \bii\b, \biii\b and so on for 12 months. Manually fix
2. End dates: Search for eg 1926-1927, or 7-12 March. \d\d-\d\d, and then \b\d-\d, and so on. Manually fix
3. In **eventDate**, search for "-\d\d\d" "\b\d-" "\b\d-"

## Result

The resulting BVP data file 'BVP treehoppers repaired - dates xx', now contains extra columns:

| column | comment |
|---|---|
| eventDateBegin | YYYY-MM-DD, or YYYY-MM-, or YYYY |
| eventDateEnd | YYYY-MM-DD, or YYYY-MM-, or YYYY |

## Hyperlinks

A URL will generally be created in this DATES section, see above.

Manual creation of a hyperlink
The CSV file output from BVP has a column: **TaskID**
In Excel, create a new column called **occurrenceID**
=CONCATENATE("http://volunteer.ala.org.au/validate/task/", B2)
Duplicate all the way down. This will create a list of URLs without hyperlinks.

In Excel to give a hyperlink to a URL use this macro.
From: http://www.niallflynn.com/random-news/convert-urls-to-clickable-links-in-excel/
This macro is saved in the folder 'Excel macros'. File = hyperlinks.bas

```
Public Sub Convert_To_Hyperlinks()
Dim Cell As Range
For Each Cell In Intersect(Selection, ActiveSheet.UsedRange)
If Cell <> "" Then
ActiveSheet.Hyperlinks.Add Cell, Cell.Value
End If
Next
End Sub

Sub removeHypers()
Intersect(Selection, ActiveSheet.UsedRange).Hyperlinks.Delete
End Sub
```

**Creating the Macro**

- Open your Excel doc
- Open the macro editor by pressing ALT+F11.
- In the Tools Menu, left-click View and select Project Explorer.
- Look for the folder called 'Modules' (or just the active folder will do), right-click it, select 'Insert', then select 'Module'.
- Paste the above code into the project module you have selected.
- Press ALT+F11 to return to your Excel workbook (or click on its icon in the Windows taskbar).

**Run the Macro**

- To execute the macro, select the unclickable text links you want to convert to clickable hyperlinks.
- Press ALT+F8 to open the Macro selector window and click on the macro you just created.
- Your Links are now all Clickable! Saving you time and data entry fatigue :)

This macro is saved in the folder 'Excel macros'. File = hyperlinks.bas

**Import existing Macro**

Hyperlinks Macro sits in a directory: 'Excel macros - use alt-F11'

- Open your Excel doc
- Open the macro editor by pressing ALT+F11.
- In the File Menu, select Import File...
- Look for the folder called 'hyperlinks.bas' . Click and open it.
- Press ALT+F11 to return to your Excel workbook (or click on its icon in the Windows taskbar).

# Chapter 3 Methods

A METHOD is needed to define an event.

## Actions

1. Create a reference picklist for METHODS
2. Clean data exported from BVP
3. Pass through to EVENTS process to be included with PARTIES, SITES, and DATES

## Files used

| Filename | Dependency | Comment |
|---|---|---|
| BVP treehoppers repaired - dates xx | From BVP with DATES repaired | Input file |
| EVENTS ento 51k with counts methods xx | From EMu – all ento events | Input file |
| Event methods changelist xx | JSON M-XL lookup methods | Reference |
| METHODS_PL reference xx | JSON V-ML lookup method picklist | Output file & Reference |
| BVP treehoppers repaired - dates+methods xx | Action file for EVENTS procedure | Output file |

## Create a METHODS reference picklist – use OpenRefine

### Preliminary – lookup table

OpenRefine can make use of a lookup table.
Create a spreadsheet lookup table. It will need at least two columns: one labelled 'bad' one labelled 'good'. This can be used to substitute improved values without needing to write a rule. eg bad = 'M.V. Lamp', good = 'MV lamp'.

Open this spreadsheet in OpenRefine, and name it: 'Event *methods* changelist xx'

In the JSON script 'M methods' the first process calls this lookup table, corrects misspellings, removes inappropriate content and standardises names of methods.

### Process to create a METHODS reference picklist

See the MSWord doc: 'Regular expressions for events xx' for the detailed GREL commands

**Input file**: 'EVENTS ento 51k xx'. Create a project in OpenRefine
**Procedure**: z-stack-create-events-methods-reference-picklist.json (same as for events-methods below)

This set of procedures does the following:

1. Apply the following JSON script:
    a. JSON M methods
    b. JSON M-XL lookup methods (uses 'Event methods changelist xx')
    c. JSON V-ML lookup method picklist (uses METHODS_PL reference – as a check. Puts an 'x' if there is an unknown method)

Manually check for a method='x'. Fix manually in 'METHODS_PL reference xx' and re-run.

First time, or to re-create a new reference picklist from scratch

**Export** as Excel file and save as 'METHODS_PL reference xx'.

Use Excel

1. Remove all columns except **ColCollectionMethod**
2. Re-import this spreadsheet as 'METHODS_PL reference xx' into OpenRefine

**'METHODS_PL reference xx'** is now the reference picklist.

'METHODS_PL reference xx' is a file with one column

| column | comment |
|---|---|
| Method | All methods in EMu after cleaning |

## Clean BVP methods data and compare against reference picklist

### Preliminary

See Chapter 1 Preliminary preparation for manual preparation of spreadsheet related to METHODS.

The following changes were made to spreadsheet column headings 'BVP treehoppers repaired xx'

| original heading | change to | comment |
|---|---|---|
| samplingProtocol | samplingProtocol-original | copy, rename and move to end |
| samplingProtocol | ColCollectionMethod | rename |

### Check against METHODS picklist

The same cleaning procedures (M methods and M-XL lookup) as used for the reference picklist above is used on the BVP export data.

**Process to check against METHODS picklist**

See the MSWord doc: 'Regular expressions for events xx' for the detailed GREL commands

**Input file**: 'BVP treehoppers repaired - dates xx' . Create a project in OpenRefine

**Procedure**: z-stack-check-against-methods-picklist.json

This set of procedures does the following:

1. Apply the following JSON scripts:
   a. JSON M methods
   b. JSON M-XL lookup methods (uses 'Event methods changelist xx')
   c. JSON V-ML lookup method picklist (uses 'METHODS_PL reference xx')
   d. JSON M-SS strip and synonymise methods (UV light = at light)

**Manually** check the column: ColCollectionMethod

An 'x' will be here if there is no match with an existing method. Either check the spreadsheet and re-run, or make an exception in the changelist file 'Event methods changelist xx'

Note: For a good catch-all use: 'Active Sampling'

**Export** as Excel file and save as "BVP treehoppers repaired - dates+methods xx"

## Result

The resulting BVP data file 'BVP treehoppers repaired - dates+methods xx' now contains these columns:

| column | comment |
|---|---|
| samplingProtocol-original | Methods from BVP |
| ColCollectionMethod | Cleaned up and checked methods |
| strippedMethod | fingerprint of method with synonymy (eg UV light = MV lamp) |

# Chapter 4 Parties

People are known as PARTIES. Sometimes they are referred to as NAMES.

## Actions

1. Create a reference picklist for PARTIES
2. Clean data exported from BVP
3. Compare exported data to picklist and check for existing parties IRN

## Files used

| Filename | Dependency | Comment |
|---|---|---|
| BVP treehoppers repaired - dates xx | From BVP with DATES repaired | Input file |
| PARTIES-ento 7k | From EMu | Input file |
| PARTIES-ento full+brief names xx | Modified 'Parties – entomology 7k' | Input file |
| Event parties changelist xx | JSON N-XL lookup names | Reference |
| PARTIES_PL ento reference xx | JSON V-NL lookup name IRN<br>JSON V-N1L lookup single name(4x)<br>JSON V-IDL lookup ID name | Output file & Reference |
| PARTIES_PL mal reference xx | JSON V-NL lookup name IRN mal<br>JSON V-N1L lookup single name mal (4x)<br>JSON V-IDL lookup ID name mal | Output file & Reference |
| BVP treehoppers - baseline dates+parties xx | Action file for SITES procedures | Output file |

## Create a PARTIES reference picklist – use OpenRefine

This procedure creates a reference picklist of PARTIES called 'PARTIES_PL ento reference xx' The picklist is derived from PARTIES in EMu, at this stage just those parties that have been used by Entomology.

- **Cleaning**. People's names are cleaned, spelling corrected, made consistent, and more readable. For example, 'Dr David Mc Alpine' has been changed to 'David McAlpine'
- **Sorted**. Improper names, or those of ships are removed. References to names of two or more people are kept for BVP1 (first iteration of BVP), but will be removed to create a picklist for BVP2.
- **Clustering**. Using OpenRefine, the picklist is minimised and some spellings corrected
- **Clean-up**. Dodgy spellings removed.

The picklist created has a column called **strippedFullName**, which is a fingerprint of the person's name.

### Preliminary 1 – prepare spreadsheet of Parties

Run a dump of the PARTIES used by Department = Entomology

EMu

Parties, TAB=Security, Department=Entomology

Search, Reports, John Tann Picklist, Report All
Save in Parties/Original directory as 'ento_unique_parties_14-08-13.xlsx'
This spreadsheet holds about 7,000 entomology parties.

**Action:**
Open spreadsheet in Excel. Save as 'Parties-ento xx'
Add two columns:

1. sequence — helpful for tracking
2. NamFullName-original — a copy of NamFullName

'Parties-ento baseline xx' spreadsheet

| Column heading | comment | Action |
|---|---|---|
| sequence | reference | Create |
| party_irn | EMu | |
| NamFullName | EMu | |
| NamBriefName | EMu | |
| CountOfUse | Number of uses in EMu | |
| NamFullName-original | Copy of NamFullName | Create |

## Preliminary 2 – including brief names, surnames and initial initials

EMu uses a brief name as well as a full name. These brief names can be used as a reference in two ways:

1. As a match for a name on a label (eg label says 'D.K. McAlpine', EMu full name is 'David K. McAlpine' and EMu brief name is 'D.K. McAlpine'. In this case the brief name helps with matching the label.)
2. As a cross reference. For example, event summary data uses brief names.

**Process to include brief names**
Add 'NamBriefName' column to the end of 'NamFullName' column in the spreadsheet of PARTIES

1. Open 'Parties-ento' in Excel
2. Copy entire table and paste into a second worksheet. Names it **brief name**
3. In second worksheet, copy contents of column 'NamBriefName' onto 'NamFullName' column. This table now holds two columns of brief names with different headings.
4. Renumber the sequence starting at 1,000,001
5. Copy this entire second worksheet and paste onto the end of the first worksheet, removing second set of column headings.
6. Save as 'PARTIES-ento full+brief names xx'

**Process to include surnames**
Add 'surnames to the end of 'NamFullName' column in the spreadsheet of PARTIES
Copy second worksheet from brief name exercise, above. Name it **surname**

1. Copy and paste **NamBriefName** to last column in worksheet
2. Rename to **BriefNameTreated**
3. Find
   *,*
   *&*
   * and *
   * et al*

*collection*
*ex.*
and replace with “”

4. Ensure that initials are separated from surname by a space
   Replace ‘.’ with ‘. ‘ (replace ‘dot’ with ‘dot-space’
   Replace 2-spaces with 1-space. Repeat this bit.
5. Copy and paste this column **BriefNameTreated** to last column in worksheet. The last two columns in worksheet are now identical.
6. Rename to **surname.**
7. Find (get rid of anything before the surname)
   ‘*. ‘ ie anything before and including the last dot and space
   *.* ie anything with a dot in it
8. Now obsolete
   a. In column **NamBriefName** find and replace ‘. ‘ with ‘. |’
   b. Data / Text to columns / Delimited / Other = ‘|’
   c. Give name to new column: **surname.** Some of these surnames will have ‘&’, etc don’t worry, these will disappear
   d. Give name to next columns ‘**x1’** to **‘x10’**. Sort on it
   e. Delete surnames and all columns to the right where there is something in column **‘x’**
   f. Find ‘*.*’, ‘*&*’, ‘* and *’, ‘* et al*’, *collection* and replace with “”
9. Copy contents of column: **surname** onto ‘**NamFullName’** column. This worksheet now holds two columns of names one with surnames and one with brief names.
10. Renumber the sequence starting at 2,000,001
11. Copy this second worksheet – up to and including NamFullName-original and paste onto the end of the first worksheet, removing second set of column headings.
12. Save as ’PARTIES-ento full+brief+sur names xx’

**Process to include initial initials**

Add ‘single-initial’ to the end of ‘**NamFullName’** column in the spreadsheet of PARTIES
Copy third worksheet from surname exercise, above. Name it **1-initial**

1. Copy column **BriefNameTreated** and paste on end
2. Rename to **1-initial**
3. In column **1-initial** find and replace ‘.* ‘ with ‘. ’ (replace dot-star-space with dot-space)
4. In column **1-initial** find and replace ‘. * ‘ with ‘. ’ (replace dot-space-star-space with dot-space). Repeat until no more
5. Copy contents of column: **1-initial** onto ‘**NamFullName’** column. This worksheet now holds two columns of names one with an initial initial and one with brief names.
6. Renumber the sequence starting at 3000001
7. Copy this worksheet – up to and including NamFullName-original and paste onto the end of the first worksheet, removing second set of column headings.
8. Save as ’PARTIES-ento full+brief+sur+1init names xx’

**Ensure IRNs are text**

May not be needed now (July 2013)
IRNs are numbers, but Excel scrambles them.
This works from home (Excel 2003):

- Rename column = “irn”, to column= “irn-original”
- New column
  =TEXT(B2, “0”)

- double click and fill the column with this formula
- copy and paste-special this column back on to column **irn** (paste as VALUES)
    - don't bother cutting and pasting as "values", as this works

This worked at AM – but it's clunky and leaves IRN as a non-number – eg "irn12345":

- Create an extra column for IRNs
- =CONCATENATE("irn", A2) – this should create text something like: "irn123456"
- double click and fill the column with this formula
- copy the VALUE of this column to another column, label it "**irn**"

## Preliminary 3 – lookup table of dodgy people names

Note. This is no longer used. See Preliminary 4 below

OpenRefine can make use of a lookup table.
Create a spreadsheet lookup table with at least two columns:

1. one column: 'irn'
2. another column called 'dodgy' with an 'x' in it

Put into this spreadsheet any name that you come across that shouldn't be used – misspellings, numbers, inappropriate characters, etc.

**Ensure IRNs are text**
IRNs are numbers, but Excel scrambles them.
This works from home (Excel 2003):

- Rename column = "irn", to column= "irn-original"
- New column
- =TEXT(b2, "general") in new column
- double click and fill the column with this formula
- don't bother cutting and pasting as "values", as this works
- Do the same for the "PARTIES-ento full+brief names xx" above

This worked at AM but is clunky and leaves IRN as a non-number – eg "irn12345":

- Create an extra column for IRNs
- =CONCATENATE("irn", A2) – this should create the text something like: "irn123456"
- double click and fill the column with this formula
- copy the VALUE of this column to another column, label it "**irn**"

This didn't work

- Create an extra column for IRNs
- =value(A2)  convert to a number and repeat for entire column
- Copy and paste values

Save the spreadsheet as 'Parties blacklist xx'
where xx is the latest version. Note: do not repeat version names.

Open this spreadsheet in OpenRefine, and name it: 'Parties blacklist xx'
In the JSON script 'N-BL' the first process calls this lookup table, and rejects the misspellings and other dodgy people names. *You will need to adjust this line in the JSON 'N-BL' script*.

## Preliminary 4 – lookup table

OpenRefine makes use of a lookup table.

Create a spreadsheet lookup table. It will need at least two columns: one labelled 'bad', one labelled 'good'. This can be used to substitute improved values without needing to write a rule. eg bad = 'Bickle' good = 'Bickel'
For super dodgy names, use bad = 'mclapin', good ='z' . where it gets given a dummy value.

Open spreadsheet in OpenRefine, and name it: 'Event parties changelist xx'
A JSON script, labelled something like 'JSON N-XL lookup names xx', calls its lookup table, and corrects misspellings and brackets.

## Process to create a PARTIES reference picklist

See the MSWord doc: 'Regular expressions for parties' for the detailed GREL commands

**Input file**: 'PARTIES-ento full+brief+sur names'. Create a project in OpenRefine
**Procedure**: z-stack-create-parties-reference-picklist.json

This set of procedures does the following:
Work first with brief names

1. Apply the JSON script
    a. N pre-NA-B rename BriefName column
    This script re-names column: "NamBriefName" to "workingName"
2. Apply the following JSON scripts:
    a. NA cleaning names - general
    b. NB cleaning names - specific
    c. do not run NC – we do not want to get rid of any names yet.
    d. ND name spelling and clusters
        i. N-XL lookup names (no longer used – causes OpenRefine problems)
    e. N-FS fix spelling names
3. Apply the following JSON script
    NR remove columns – gets rid of scaffolding
    This script removes the following columns:
        a. spellCheck
        b. badSpelling
        c. NameFullNamePreSort
    JSON N-post-NR-B replace BriefName
        a. Return column names: "workingName" to "NamBriefName"

Then work with full names
Note: Two references to one changelist (one above here and one below) may give dodgy results. I may need to split into two procedures for consistency, which I would like to avoid if possible, as it is another layer of work.

4. Apply the JSON script
    a. N pre-NA-F rename FullName column
    This script re-names column: "NamFullName" to "workingName"
5. Apply the following JSON scripts:
    a. NA cleaning names general
    b. NB cleaning names specific
    c. ND name spelling and clusters
    d. N-1 on single names
    e. N-FS fix spelling names

f. N-XL lookup names (uses 'Event parties changelist xx')
g. NC presenting changes to names

6. Apply the following JSON script
   NR remove columns – gets rid of scaffolding
   This script removes the following columns:
   a. spellCheck
   b. badSpelling
   c. NameFullNamePreSort
   JSON N-post-NR-F replace FullName
   a. Return column names: "workingName" to "NamFullName"

**Export** as Excel file and save as 'PARTIES_PL ento reference xx'.
Use Excel

1. Remove: preferredFullName = N, misspelt
2. Remove rows with blank **NamFullName** they are all misspelt, or something
3. Remove rows with **NamFullName = z,** they are dodgy
4. Remove rows with blank **NamBriefName** if they are dodgy
5. Strip out unneeded columns. Keep these columns:
   party_irn
   NamFullName
   strippedFullName
   NamBriefName
   OK to include, but really for your reference
   CountOfUse
   NamFullName-original
6. Re-import this spreadsheet as 'PARTIES_PL ento reference xx' into OpenRefine

'**PARTIES_PL ento reference xx'** is now the reference picklist for PARTIES.

Note: For the early BVP data, multiple collectors for one event were all lumped together. For later data these have been separated. This means that the reference list for checking against the early BVP data needs to include multiple people.

## Clean BVP parties data and compare against reference picklist

This procedure does two things:

1. clean up the parties info in a file exported from the Biodiversity Volunteer Portal
2. compare that PARTIES data with the PARTIES reference picklist above

The cleaned and referenced PARTIES are also used to create an EVENT.

### Prepare BVP parties data with Excel

BVP data comes as a series of spreadsheets – one for each expedition. In total there are about 11,000 entomology records. Note that these are the same spreadsheets used for SITES.
**Filename**: Example 'BVP treehoppers xx' or 'BVP treehoppers repaired - dates xx'

Change spreadsheet column headings

| original heading | change to | comment |
|---|---|---|
| | | |

Global replace all '\n' with a space.
Fix any obvious spelling and other errors – eg things in wrong columns

Save spreadsheet as:
'BVP treehoppers repaired xx' - where xx is a sequence marker

## Clean and compare BVP parties data using OpenRefine

## Check against picklist

The same cleaning procedures 'NA' and 'NB' as used for the reference picklist above are used on the BVP export data.

## Process to check against PARTIES picklist

See the MSWord doc: 'Regular expressions for parties' for the detailed GREL commands

**Input file**: 'BVP treehoppers repaired – dates+methods xx' . Create a project in OpenRefine
**Procedure**: z-stack-check-against-parties-picklist-BVP-identifiedBy.json

This set of procedures does the following:
See also 'NE' in 'Regular expressions for parties' for how to do this
Work first with identifiedBy names

1. Apply the JSON script
    a. V-N pre-NA-I rename identifiedBy column
    This script re-names column: "identifiedBy" to "workingName"
    (repeat this later for "recordedBy" see 'NE' about how to do this)
2. Apply the following JSON scripts:
    a. NA cleaning names - general
    b. NB cleaning names - specific
    c. N-XL lookup names (uses 'Event parties changelist xx')
    d. N-XD name spelling and clusters
    e. NS stripped names (removes 'and' and numbers)
        1. this step has been moved further down (and re-named V-IDL) with combined idBy and recordedBy
        2. V-NL lookup name IRN (uses 'PARTIES_PL reference xx**'**)

Note: These JSON scripts will be applied to several columns:
identifiedBy
recordedBy
Note 1: 'V-NL' looks up an IRN for the name. It needs to be run after 'N-XD'. Note that the filename of the reference picklist, "**PARTIES_PL ento reference xx"**, is held inside 'V-NL'. When the version number of the reference changes, this name will need to be changed in 'V-NL'.
Note 2: 'N-XL' refers to a good/bad name changelist, where spellings and other one-off errors are corrected. Changelist filename: "**Event parties changelist xx**". When the version number of the reference changes, this name will need to be changed in 'N-XL'

3. Apply the JSON script
    V-N-post-NL-I replace identifiedBy
    This script will:
    a. Re-name column: "name_irn" to "identifiedBy_irn"

b. Return column name: “workingName” to “identifiedBy“

Export as Excel file
Give it a name “BVP treehoppers repaired - dates+methods+identifiedBy xx”
This is a needed intermediate step as unfortunately OpenRefine can only do one lookup at a time

I think? I have overcome this by running all the lookups together.

**Input file**: ‘BVP treehoppers repaired - dates+methods+identifiedBy xx. Create a project in OpenRefine
**Procedure**: z-stack-check-against-parties-picklist-BVP-recordedBy.json
Then work with recordedBy names

4. Apply the JSON script
   a. V-N pre-NA-R rename recordedBy column
   This script re-names column: “recordedBy” to “workingName”
5. Apply the following JSON scripts:
   a. NA cleaning names - general
   b. NB cleaning names - specific
   c. N-XL lookup names (uses ‘Event parties changelist xx’)
   d. N-XD name spelling and clusters
   e. N-XSI strip initials
   f. V-N1S split into single names
      - Adds 8 more columns, 2 for each of 4 possible collectors
        1. collector1
        2. collector1_irn
        3. and so on
   g. V-N1L lookup single name
6. Apply the JSON script
   a. V-IDL lookup ID name
      - moved here from first section. It seems to work if run at the same time as the other lookups (ie without changing contents fo the column, and then doing a re-lookup, or something)
   b. V-N-post-NL-R replace recordedBy
      This script will:
      - Return column name: “workingName” to “recordedBy“
      - Add two columns:
        i. strippedName (without people’s initials – for events)
        ii.

**Export** as Excel file and save as “BVP treehoppers repaired – dates+methods+parties xx”

“**BVP treehoppers repaired – dates+methods+parties xx**” is the BVP data file with extra columns due to parties:

sequence (for PARTIES, and SITES)
identifiedBy_irn
recordedBy_irn (from PARTIES)
recordedByBriefName
strippedName (for EVENTS)
collector1
collector1_irn

collector2
collector2_irn
collector3
collector3_irn
collector4
collector4_irn

## Result

The resulting BVP data file “BVP treehoppers repaired – dates+methods+parties” contains extra columns:

| column | comment |
|---|---|
| identifiedBy_irn | EMu party IRN |
| recordedBy_irn | EMu party IRN |
| recordedByBriefName | recordedBy written briefly |
| strippedName | fingerprint of brief name of recordedBy |
| collector1 | collector, or first person of a group |
| collector1_irn | EMu party IRN |
| collector2 | second collector of a group |
| collector2_irn | EMu party IRN |
| collector3 | third collector of a group |
| collector3_irn | EMu party IRN |
| collector4 | fourth collector of a group |
| collector4_irn | EMu party IRN |

# Chapter 5 Sites

## Actions

1. Create a reference picklist for SITES
2. Clean data exported from BVP
3. Compare exported data to picklist and check for existing site IRN

## Files used

| Filename | Dependency | Comment |
|---|---|---|
| BVP treehoppers repaired - dates+parties xx | From BVP with DATES+PARTIES repaired | Input file |
| SITES-ento baseline xx | From EMu | Input file |
| Event sites changelist xx | JSON P-XL lookup places | Reference |
| SITES_PL ento reference xx | JSON VP-PL lookup place IRN | Output file & Reference |
| BVP treehoppers repaired - dates+parties+sites xx | Action file for METHODS procedure | Output file |

## Create a SITES reference picklist – use OpenRefine

This procedure will create a reference picklist of SITES called 'SITES_PL ento reference xx' The picklist is derived from SITES in EMu. A different picklist is created for each department – the SITES picklist for Entomologoy is different to the SITES picklist for Malacology.

- **Cleaning**. Locations will be cleaned, made consistent and more readable. For example, '1.5kms (about 2.5 mi) nth of t-off' will be changed to '1.5 km N of Turnoff'
- **Clustering**. Using OpenRefine, the picklist will be minimised and some spellings corrected
- **Presenting**. Extra columns will be added
- **Geography**. A check will be made to confirm that each record exists within or close to its state boundary

The picklist will have a column called **strippedLocation**, which is a fingerprint of the Location with most prepositions (at, in, just, to …) removed. Exception: the word NEAR has been retained – I have assumed that, for example, 'near Bondi' is not the same as 'Bondi'. *Though 'Bondi, near Sydney' is probably the same as 'Bondi, Sydney'.*

The file from EMu comes from a report generating script developed by Mikey.

Entomology: About 20,000 records.
Malacology: About 75,000 records.

### Preliminary 1 – sites spreadsheet

Run a dump of the SITES used by Department = Entomology (or another department)

EMu
Sites, TAB=Security, Department=Entomology
Search, Reports, John Tann Picklist, Report All

Save in Sites/original/ as ‘unique sites ento xx’

**Action:**
Open spreadsheet in Excel.
Add two columns:

1. sequence — helpful for tracking
2. LocPreciseLocation-original — a copy of LocPreciseLocation

Save as ‘Sites-ento baseline xx’. Include these fields:

| column | check | comment |
|---|---|---|
| sequence | | add this |
| site_irn | | |
| LocOcean | | |
| LocCountry | delete blanks if dodgy | |
| LocProvinceStateTerritory | delete blanks if dodgy | |
| LocDistrictCountyShire | | |
| LocTownship | | |
| LocPreciseLocation | | |
| CountOfUse | | |
| Elevation_metres | | |
| Elevation_feet | | |
| dLat | | |
| dLong | | |
| DMSLat | | |
| DMSLong | | |
| SiteNumber | | |
| LocPreciseLocation-original | | Create. Copy from LocPreciseLocation |

Checks

- site_irn – delete any blanks

If **LocOcean** = blank and **LocCountry =** blank and **LocProvinceStateTerritory** = blank and **LocDistrictCountyShire** = blank and **LocTownship** = blank and **LocPreciseLocation** = blank, then delete record

- If **LocProvinceStateTerritory** = blank and **LocDistrictCountyShire** = blank, then check. If dodgy, then delete.
- If **CountOfUse** = BLANK, then make '0'. (It had been allocated, but not yet used)
- **LocOcean**: Check consistency – eg just Pacific Ocean (remove South- North-, etc)

Re-sequence and save as ‘SITES-ento baseline ok xx’
Load into OpenRefine.

**Note these checks are now done within OpenRefine – (used to be done in Excel)**

- Add Country = Australia for obvious places
- Add states for Australia
- Change case of Countries, Provinces, Districts
    - sort on LocCountry
    - create a new column to the right of LocCountry
    - EITHER

        - Formula =Proper(C3) – only do this for the cells with things in them (not blanks). Note that this formula changes the first letter of all the words in a Country's name to Uppercase including United States Of America. ie capital 'Of'. This is fixed in JASON PB.
        - OR use Data / Flash Fill after showing how to do it… (doesn't always work)
    - Copy values back to original column
    - Delete working column
- Quick check of **LocCountry** = Australia, and spelling of states in **LocProvinceStateTerritory**.
- Old country names are transposed to their modern equivalent in JSON PB, for example British New Guinea is now Papua New Guinea.
- If needed, fix uppercase. If errors are later found, fix in 'SITES-ento complete pre-sort xx'

## Preliminary 2 – lookup table

OpenRefine can make use of a lookup table.
Create a spreadsheet lookup table with at least two columns of locations: one labelled 'bad' one labelled 'good'. This can be used to substitute improved values without needing to write a rule. eg bad = 'Sydeny' good = 'Sydney'

Open this spreadsheet in OpenRefine, and name it: 'Event sites changelist xx'
In the JSON script 'PD' the first process calls this lookup table, and corrects misspellings and brackets.

## Preliminary 3 – including locations dropping Townships or Districts

This preliminary procedure adds to the reference list of locations, a set of locations with Townships and Districts removed.

EMu uses Township and District to help define a place.
Example: Bayview, NSW does not exist, but 'Bayview + Township=Sydney + State=NSW' does exist.

**Process to include locations without Township or District**

1. Open 'SITES-ento baseline xx' in Excel
2. Copy entire table and paste into a second worksheet. Label it 'second'.
3. In second worksheet, throw away stuff:
    a. Sort on LocDistrictCountyShire
    b. Sort on LocTownship
    c. Chop off spreadsheet below the bottom of these sorted columns. ie throw away all rows that have nothing in either Township or District
    d. Sort on LocPreciseLocation
    e. Chop off spreadsheet below the bottom of this sorted column. ie throw away empty locations
    f. Delete contents of LocDistrictCountyShire and LocTownship (keep headers)
    g. This is a set of locations with District and Township removed (about 1000)
4. Sort on sequence
5. Renumber the sequence by adding 1,000,000 …
    a. duplicate column A -> column B

b. formula =(a2+1000000)
c. copy values back to column A
d. delete column B

6. Copy this entire second worksheet and paste onto the end of the first worksheet.

More to go: We have a problem with District = Blue Mountains, Township=Blackheath – we miss Blackheath, NSW

7. Copy entire table and paste into a third worksheet. Label it 'third'.
8. In third worksheet throw away everything except those with a District AND Township AND empty Location
    a. Sort on LocPreciseLocation
    b. Chop off all those with a LocPreciseLocation
    c. Sort on LocTownship
    d. Chop off all those without a LocTownship
    e. Sort on LocDistrictCountyShire
    f. Chop off all those without a LocDistrictCountyShire
    g. Delete contents of LocDistrictCountyShire (keep header)
    h. This is a set of Townships without precise location and with districts (about 30)
9. Renumber the sequence by adding 2,000,000 …
    a. duplicate column A -> column B
    b. formula =(a2+2000000)
    c. copy values back to column A
    d. delete column B
10. Copy this entire second worksheet and paste onto the end of the first worksheet.

For Malacology, these above two procedures added another 40 rows.

## Elevation

Elevations are stored in EMu as both metres and feet.
Label data comes with elevation as either metres or feet.
As a way of marrying the two, duplicate those records with elevation, and have two elevations for each site – one for metres and one for feet, though without any reference to feet or metres.

When checking a site on a label for a match in the picklist, an elevation will match one of either the metre value or the feet value, either is okay as they both have the same SITE IRN.

**In the compound worksheet**

Duplicate the column **Elevation_metres** and paste in the column before. Rename new column **Elevation_NoUnits.** Give it a colour, eg yellow.
Copy the worksheet and paste into a new worksheet. Label it **elevation.**
In the **elevation** worksheet

1. In **elevation** worksheet throw away everything except those with an elevation
    a. Sort on Elevation_NoUnits
    b. Chop off all those without an Elevation
    c. Delete all those with an Elevation=0, or silly
    d. Copy column Elevation_feet on top of column Elevation_NoUnits.
    e. Renumber the sequence by adding 3,000,000 …
        i. duplicate column A -> column B
        ii. formula =(a2+3000000)

iii. copy values back to column A
iv. delete column B

f. Colour entire worksheet green. (About 4,000 records for SITES)

2. Copy this entire **elevation** worksheet and paste onto the end of the first worksheet. We now have a worksheet with a column labelled **Elevation_NoUnits.** This column contains the elevation twice - in both metres and feet - for each SITE with an elevation.

## Synonymy

This section works for Entomology (~25k sites), but is too big for Malacology (74k sites)

For Malacology
Run through the OpenRefine steps (below) and select the preferred records (preferredLocation = Y).
Save as 'SITES-mal first sort xx.xls'
Then duplicate the entire worksheet (shown below)

And then, after you've done that, remove the columns added by OpenRefine and re-run through the OpenRefine steps again, to create a set with phrases like ", near Dubbo" removed from the duplicate.
ie Remove these columns

strippedLocation
PreferredLocation
strippedLoc-D
alteredLcocation
facetLocation
LocationNoNear
strippedLoc-NoNear
OutOfBounds

To determine synonyms, create a second set of SITES to fiddle with.
This will be a set where 'near Dubbo, or 'Bondi, near Sydney' is removed.
It is a marked duplication of the above compound spreadsheet, tacked onto the bottom of the above spreadsheet.

**Duplicate the entire compound worksheet above**

Create a column: "**duplicate**" after **CountOfUse**. Leave column empty.
Copy the compound worksheet and paste into a new worksheet. Label it **duplicates**
In the **duplicates** worksheet

1. In column **CountOfUse**, give these events a low priority. They are really only pointers for other events
    a. Subtract 2000: due complications when all other locations=0
    Copy column=CountOfUse. In copied column use formula =(L2-2000)
    *** Care make sure that all cells copied ***
    Copy and paste special = values back onto original
    Delete copied column
2. In the column: **duplicate** (after **CountOfUse)**: Fill it with the word 'duplicate'
3. Renumber **sequence** by adding 6,000,000

a. Copy column= **sequence**. In copied column use formula =(A2+6000000)
   Copy and paste special = values back onto original
   Delete copied column
4. Copy entire worksheet and paste onto the bottom of the compound worksheet.
5. Save as 'SITES-ento baseline+duplicates xx'

There may be a few dodgy entries like "Public Jetty + State = NSW", but they seem small and less likely to get a hit.

## Process to create a SITES reference picklist

See the MSWord doc: 'Regular expressions for sites xx' for the detailed GREL commands

**Input file**: 'SITES-ento baseline+duplicates xx'. Create a project in OpenRefine
**Procedure**: z-stack-create-sites-reference-picklist.json (Parts 1&2)

This set of procedures does the following:

1. Apply the following JSON scripts:
   a. PA cleaning places general
   b. PB cleaning places specific
   c. PMF manually fix
   d. P-XL lookup places (uses 'Event sites changelist xx')
   e. PD spelling cluster-edits corrections
   f. PG geography
   g. P-SS2 strip site – separates synonymy
   h. P-RSS-S reconcile strip site for SITES
   i. PC presenting changes to places
   j. PC+ after PC for SITES
   k. PS sort SITES

   Note: 'PC' is the presentation module which needs to be run after 'PB', 'PD' and 'PG'
2. Optionally clean up with the following JSON script:
   l. PR remove columns (OK here – sorting is done)

## Once-off check for brackets and miles-metres fix: DONE

Checking for brackets and fixing for miles and metres has been done. The results are fed into the reference file 'Event sites changelist xx'. The next two sections are included here for reference.

## Check for brackets: DONE

Brackets are a pain. Convert them to something else.

1. Stop the process at the point where the 'changelist' is looked up (about step 500). Step a couple of steps after that, where the brackets around km and miles are removed.

2. In column **LocPreciseLocation**
   Filter = "(" or ")"  (ie brackets)
3. Extract the two columns
4. Open the spreadsheet 'Event sites changelist xx.xls'
5. Create a new worksheet, paste the two columns. Rename the worksheet "SITES-ento xx".
6. Create an extra column 'good'

7. For each record with a bracket, put the fixed version in column 'good'.
8. Paste all fixed rows into the primary worksheet 'Events sites changelist xx'

Either check for miles-metres irregularities now, or do the following three steps

9. Open 'Events sites changelist xx' in OpenRefine
10. Update reference to 'Events sites changelist xx' in JSON P-XL lookup places
11. Re-run the process to create SITES reference picklist

## Check for miles-metres irregularities: DONE

'm' is used for both miles and metres. Sorting that out is done mostly by script, but for distances between 10 and 100, metres and miles are best sorted by human.

1. In Undo/Redo, filter on miles-metres
2. Stop the process at the last reference to "miles-metres", and facet.
3. In column: **miles-metres**
   Facet on BLANK = FALSE
4. In column: **LocPreciseLocation**
   Text facet

5. Extract the two columns
6. Open the spreadsheet 'Event sites changelist xx.xls'
7. Create a new worksheet, paste the two columns. Rename the worksheet "SITES-ento xx".
8. Create an extra column 'good'
9. Where a location has a wrong reference as 'metres' instead of miles, put the fixed version in column 'good'.
10. Paste all fixed rows into the primary worksheet 'Events sites changelist xx'
11. Open 'Events sites changelist xx' in OpenRefine
12. Update reference to 'Events sites changelist xx' in JSON P-XL lookup places
13. Re-run the process to create SITES reference picklist

## After processing in OpenRefine

**Export** as Excel file and save as 'SITES_PL ento reference xx'.

The column **alteredLocation** (about column "N") contains descriptors for what has been changed in the location

| alteredLocation | example | note |
|---|---|---|
| removed near | ", near Coffs Harbour" | keep |
| removed Sydney | "Bondi, Sydney" | keep |
| removed Blue Mountains | "Katoomba, Blue Mountains" | keep |
| removed Township | "Brisbane" | keep |
| removed districts | x, Iluka district | keep |
| removed LocDistrictCountyShire | Christmas Island | keep |
| removed LocTownship | Orange | keep |
| removed N-S-E-W | X National Park, S of Dubbo | keep |
| removed other | Tamborine Mount | keep |
| removed island groups | x Island, Great Barrier Reef | keep |

The column **facetLocation** (about column 'M') was used as scaffolding. Not required here.

| facetLocation | example | note |
|---|---|---|
| near | "Murray River, near Echuca" | don't use here |
| extensive feature | lake, river, road | don't use here |
| extensive – road | road, street, highway, track | don't use here |

You don't have to do anything about these, however for interest, changed rows were: "removed x" such as "removed near" "removed Sydney" "removed Blue Mountains" "removed Township" "removed other"
Not used in the procedures were: "unchanged near" "extensive – road" or "extensive feature"

Use Excel

1. Cut the entire worksheet and paste into a new worksheet. Name the worksheet 'SITES_PL reference'. Move this worksheet to first position.

In second worksheet 'SITES_PL reference'

2. Remove: preferredLocation = N
3. In column: LocPreciseLocation, Check for LocPreciseLocation=x and LocPreciseLocation=z and LocPreciseLocation="" (something but not a blank- find by sorting). Remove them.
4. Strip out unneeded columns. Keep these columns in a separate worksheet, perhaps:
   - sequence (optional but useful for re-ordering)
   - site_irn
   - LocOcean
   - LocCountry
   - LocProvinceStateTerritory
   - LocDistrictCountyShire
   - LocTownship
   - LocPreciseLocation
   - strippedLocation
   - CountOfUse (this is a reference after all)
   - Elevation_NoUnits
   - Elevation_metres
   - Elevation_feet
   - dLat
   - dLong
   - DMSLat
   - DMSLong
   - SiteNumber
   - strippedSiteNumber
5. Re-import this spreadsheet as 'SITES_PL ento reference xx' into OpenRefine

'**SITES_PL ento reference xx'** is now the reference picklist.
Change the reference to **SITES_PL ento reference xx** in JASON VP-PL

## Clean BVP sites data and compare against reference picklist

This procedure does two things:

1. clean up the sites info in a file exported from the Biodiversity Volunteer Portal

2. compare that SITES data with the SITES reference picklist above

This section will be run after Dates and Parties, the input filename will be something like: 'BVP treehoppers repaired – dates+parties xx'.

This section is stand-alone. Although it does not need any other processes such as dates, parties or methods to be run before this, keeping the sequence in order is important for version control.

## Prepare BVP sites data with Excel

BVP data is downloaded as a series of CSV files.
Preliminary manual work is done before automated processes are commenced.

See Chapter 1 Preliminary preparation, for manual preparation of spreadsheet related to SITES.

The following changes were made to spreadsheet column headings 'BVP treehoppers repaired xx'

| original heading | change to | comment |
|---|---|---|
| country | LocCountry | |
| stateProvince | LocProvinceStateTerritory | |
| verbatimLocality | LocPreciseLocation | data here will be cleaned |
| verbatimLocality | verbatimLocality-original | copy of verbatimLocality |
| | LocTownship | add |
| | LocDistrictCountyShire | add – for islands |

Do not add **locality**. This is a Google attempt at the placename, and will confuse.
Use spreadsheet tools to sort the data and:

- Add **country, state**, where indicated
- Correct obvious spellings and omissions in column **LocPreciseLocation**.
- Put Lord Howe Island and any other island under the new column: **LocDistrictCountyShire**
- Put Antarctica in **LocProvinceStateTerritory** as 'Australian Antarctic Territory'

## Clean and compare BVP sites data with OpenRefine

**Check against picklist**

1. The same cleaning procedures (PA, PB) as used for the reference picklist above are used on the BVP export data.
2. A separate procedure (VP-D) is used to rip out superfluous state and country information. Some clustering is done, though the automated clustering scripts aren't as effective as a human-monitored process, and the BVP export data should probably be re-clustered to help clean it up.
3. A separate procedure (VP-C) is used to present the BVP data. This procedure also compares against the reference picklist created above.
4. A column is added to BVP export data called site_irn. This column contains the EMu site IRN matching a location in the reference picklist.

**Process to check against picklist**

See the MSWord doc: 'Regular expressions for sites' for the detailed GREL commands

**Input file**: 'BVP treehoppers repaired – dates+methods+parties xx'. Create a project in OpenRefine

**Procedure**: z-stack-check-against-sites-picklist.json

This set of procedures does the following:

1. Apply the following JSON scripts:
    a. PA cleaning places – general (same script as for reference above)
    b. VP-CD creating districts
    c. PB cleaning places – specific (same script as for reference above)
    d. P-XL lookup VP places (uses 'Event sites changelist xx')
    e. VP-D spelling cluster-edits corrections
    f. P-SS2 site synonymy
    g. VP-C presenting changes to places
    h. VP-PL lookup place IRN (uses 18x 'SITES_PL ento reference xx')
    i. VP-T align townships
    j. VP-R remove unwanted columns

    Note: VP-PL has the filename '**SITES_PL ento reference** xx' embedded 18 times in the JSON. When the version number of the reference changes, this name will need to be changed in 'VP-PL'.

    Note: P-XL has the filename '**Event sites changelist** xx' embedded in the JSON. When the version number of the changelist changes, this name will need to be changed in 'P-XL'.

**Export** as Excel file and save as "BVP treehoppers repaired – dates+methods+parties+sites xx"

'BVP treehoppers repaired – dates+methods+parties+sites xx' is the BVP export data file with four extra columns:

strippedLocation
site_irn
LocTownship
LocDistrictCountyShire
LocPreciseLocation (cleaned up version of verbatimLocality)

**Checks**

I had trouble with OpenRefine checking against the reference. It gave inconsistent results. Sometimes it would return about 500 matching IRNs and other times 800 matching IRNs. Things that help:

- Restart OpenRefine. No need to close the project.
- This is a known bug and has been fixed in a more recent upgrade of OpenRefine. Only thing – I can't find the upgrade, and it's not a dummy-friendly site – yet. The problem is about having too many instances of lookup open at once. Don't try and lookup more than one file, or for more than too many (one?) columns.

## Manually check sites

Humans can match sites probably better than a machine.
After the machine has had a go, take the exported spreadsheet and add in manual site irns.
Add this column: **manualSite_irn** after **sequence,** or better, after **manual_irn** if it exists.

## Result

The resulting BVP data file 'BVP treehoppers repaired – dates+methods+parties+sites xx' contains extra columns:

| column | comment |
|---|---|
| site_irn | EMu site IRN |
| LocTownship | Township as used by EMu |
| LocDistrictCountyShire | Islands as used by EMu |
| strippedLocation | fingerprint representation of location – used for comparing against reference |
| LocPreciseLocation | Cleaned up version of verbatimLocality |

# Chapter 6 Events

An EVENT is a combination of:

a DATE
a PARTY, which may be more than one collector
a SITE
a METHOD

## Actions

1. Create a reference picklist for EVENTS
2. Using DATE, PARTY, SITE and METHOD data exported from BVP, create an EVENT
3. Compare exported data to events picklist and check for existing event IRN

## Files used

| Filename | Dependency | Comment |
|---|---|---|
| BVP treehoppers – baseline+dates+ methods+parties+sites xx | From BVP with DATES+PARTIES+SITES+METHODS repaired | Input file |
| COLLECTION_EVENT_PL_ALL | From EMu – all events | Input file |
| EVENTS ento baseline counts+methods | From EMu – all ento events derived from COLLECTION_EVENT_PL_ALL | Input file |
| Event dates changelist xx | JSON D-XL lookup dates in events | Reference |
| Event parties changelist xx | JSON N-XL lookup names | Reference |
| Event sites changelist xx | JSON P-XL lookup places | Reference |
| Event methods changelist xx | JSON M-XL lookup methods | Reference |
| EVENTS ento reference – dates | Output of events-dates<br>Input of events-methods<br>JSON V-DL lookup matching event dates | Intermediate<br><br>How many events dates match? |
| EVENTS ento reference – dates+methods | Output of events-methods<br>Input of events-parties 1 | Intermediate |
| EVENTS ento reference – dates+methods+identifiedBy | Output of events-parties 1<br>Input of events-parties 2 | Intermediate Parties part 1 |
| EVENTS ento reference – dates+methods+parties | Output of events-parties<br>Input of events-sites | Intermediate Parties part 2 |
| EVENTS ento reference – dates+methods+parties+sites | Output of events-sites<br>Input of events-events | Intermediate |
| EVENTS ento reference – dates+methods+parties+events 1 | Output of events-events 1<br>Input of events-events 2 | Intermediate Events part 1 |
| EVENTS_PL-ento reference xx | JSON V-EL lookup event IRN<br>JSON V-WOL lookup substandard event IRN | Output file & Reference |
| EVENTS_CEC reference xx | JSON V-CECL lookup collection event code IRN | Output file & Reference |
| BVP EMu upload list so far xx | JSON V-UL lookup import list | Reference |
| BVP treehoppers repaired – | File for import into EMu | Output file |

| Filename | Dependency | Comment |
|---|---|---|
| dates+methods+parties+sites+events xx | | |

## Create an EVENTS reference picklist – use OpenRefine

This procedure will create a reference picklist of EVENTS called 'EVENTS_PL reference xx' The picklist is derived from the EVENTS in EMu; at this stage just those events that have been used by Entomology.

- **Cleaning**. People's names, dates, sites and methods are cleaned, spelling corrected, and made consistent and more readable. This is done using the procedures developed for DATES, METHODS PARTIES, SITES, in previous chapters
- **Clustering**. Using OpenRefine, the picklist is minimised and some spellings corrected

The events picklist has a column called **strippedEvent**, which is a concatenation of the fingerprints of date, the parties, the site and the method.

The file from EMu created as a report, is called 'EVENTS-all original xx' or something like that. This has about 240,000 events. A subset of only the entomology events has been created 'EVENTS-ento original xx'. This file has about 80,000 entomology events.

### Preliminary 1 – events spreadsheet

Run a dump of the events used by Department = Entomology

EMu
Collection Events, TAB=Security, Department=Entomology
Search, Reports, John Tann Picklist, Report All

Include these fields:

| column | check | comment |
|---|---|---|
| sequence | | add this |
| event_irn | | |
| ColDateVisitedFrom | | |
| ColDateVisitedTo | | |
| SummaryData | | |
| ColCollectionMethod | | |
| NamBriefName | | |
| LocOcean | | for Malacology in particular |
| LocCountry | delete blanks if dodgy | |
| LocProvinceStateTerritory | delete blanks if dodgy | |
| LocDistrictCountyShire | | |
| LocTownship | | |
| LocPreciseLocation | | |
| CountOfUse | | include zero counts |
| DepthFrom_metres | | |
| DepthTo_metres | | |
| BottomDepthFrom_metres | | |
| BottomDepthTo_metres | | |
| Elevation_metres | | |
| Elevation_feet | | |

| column | check | comment |
|---|---|---|
| dLat | | |
| dLong | | |
| DMSLat | | |
| DMSLong | | |
| SiteNumber | | |
| CEC | | |
| participant_irn | | |
| LocPreciseLocation-original | | create a copy |
| SummaryData-original | | create a copy |
| ColCollectionMethod-original | | create a copy |

**Checks**
Re-sequence and save as 'EVENTS-ento baseline xx'

## Elevation

Taken exactly from SITES chapter above.
{Jeeze, I don't like this mucking about before starting.}

Elevations are stored in EMu as both metres and feet.
Label data comes with elevation as either metres or feet.
As a way of marrying the two, duplicate those records with elevation, and have two elevations for each site – one for metres and one for feet, though without any reference to feet or metres.

When checking a site on a label for a match in the picklist, an elevation will match one of either the metre value or the feet value, either is okay as they both have the same SITE IRN.

**In the original worksheet**

Duplicate the column **Elevation_metres** and paste in the column before. Rename new column **Elevation_NoUnits.** Give it a colour, eg yellow.
Copy the worksheet and paste into a new worksheet. Label it **elevation.**
In the **elevation** worksheet

3. In **elevation** worksheet throw away everything except those with an elevation
    a. Sort on Elevation_NoUnits
    b. Delete all those with an Elevation=0, or silly – don't worry-removed in OpenRefine
    c. Chop off all those <u>without</u> an Elevation
    d. Copy column Elevation_feet on top of column Elevation_NoUnits.
    e. Renumber the sequence by adding 6,000,000 …
        i. duplicate column A -> column B
        ii. formula =(a2+6000000)
        iii. copy values back to column A
        iv. delete column B
    f. Colour entire worksheet green. (About 7,000 records for EVENTS)
4. Copy this entire **elevation** worksheet and paste onto the end of the first worksheet.
        i. Ctrl+A doesn't copy to another worksheet so this is workaround
        ii. select header row
        iii. ctrl+shift+down-arrow
        iv. Copy-paste

5. Follow up: sort on Elevation_NoUnits and delete all those with an Elevation=0
6. We now have a worksheet with a column labelled **Elevation_NoUnits.** This column contains the elevation twice - in both metres and feet - for each SITE with an elevation.

**Size matters – so SPLIT**
If spreadsheet is longer than 65,000 rows, OpenRefine struggles and fails.

Break the file into two or more parts – 1-50,000; and 50,000-rest is OK, probably best sorted on date (or something to ensure that events are not duplicated in both parts), but you can break it on Country=Australia, or something else. Run both splits (partA and partB) against PARTIES and SITES procedures below. Any splitting will be brought back together after running the SITES procedure.

Save as:
EVENTS-ento baseline partA xx
EVENTS-ento baseline partB xx

Load into OpenRefine.
Quick check of **LocCountry** = Australia, and spelling of states in **LocProvinceStateTerritory**.
If needed, fix uppercase. If errors, then fix in 'EVENTS-ento 61k baseline xx'

## Preliminary 2 – lookup table

OpenRefine makes use of a lookup table.
Create four spreadsheet lookup tables. Each will need at least two columns: one labelled 'bad' one labelled 'good'. This can be used to substitute improved values without needing to write a rule. eg bad = 'Bickle' good = 'Bickel'

In turn, open each spreadsheet in OpenRefine, and name it: 'Event *yyyy* changelist xx'
A JSON script, labelled something like 'JSON xx-XL lookup xxx', calls its lookup table, and corrects misspellings and brackets. Do this for dates, parties, sites, methods.

Four lookup tables are needed.

| Lookup table | Script |
|---|---|
| Event dates changelist xx | JSON D-XL lookup dates in events |
| Event parties changelist xx | JSON N-XL lookup names |
| Event sites changelist xx | JSON P-XL lookup places |
| Event methods changelist xx | JSON M-XL lookup methods |

## Process to create an EVENTS reference picklist

This process makes use of other processes that were employed to generate the three previous picklists – PARTIES, SITES and METHODS. A DATE cleaning process is also used.

1. Create a project in OpenRefine
2. Get data from: 'EVENTS ento 51k – with counts+methods'

**Dates**
*Partially taken from Chapter 2 Dates, above*
See the MSWord doc: 'Regular expressions for events' for the detailed GREL commands

Mostly, dates that come out of EMu are well formatted. There are a few scrambled dates and these are corrected using the changelist: 'Event dates changelist xx'

**Input file**: 'EVENTS-ento 51k – with counts'
**Procedure**: z-stack-create-events-dates-reference-picklist.json
This set of procedures does the following:
1. Apply the following JSON script
    a. JSON D-XL lookup dates in events (uses 'Event dates changelist xx')

**Export** as Excel file and save as 'EVENTS-ento dates partA xx'. This is used as input to the METHODS procedure.

**Methods**

*Partially taken from Chapter 3 Methods, above*
See the MSWord doc: 'Regular expressions for events' for the detailed GREL commands

**Input file**: 'EVENTS-ento baseline dates partX xx'
**Procedure**: z-stack-create-events-methods-reference-picklist.json
This set of procedures does the following:
1. Apply the following JSON script:
    a. JSON M methods
    b. JSON M-XL lookup methods (uses 'Event methods changelist xx')
    c. JSON V-ML lookup method picklist (uses METHODS_PL reference – as a check. Puts an 'x' if there is an unknown method)
    d. JSON M-SS strip and synonymise methods (UV light = at light)
2. Extract Collection Event Codes from summary data (I know it's not a METHOD, but this is a good place to do it)
    a. JSON X-CXC create collection event code

Manually check for a method='x'. Fix manually and re-run.

**Export** as Excel file and save as 'EVENTS-ento methods partX xx'. This is used as input to the PARTIES procedure.

**Parties**

*Partially taken from Chapter 4 Parties, above*
See the MSWord doc: 'Regular expressions for parties' for the detailed GREL commands

**Input file**: 'EVENTS-ento methods partX xx'
**Procedure**: z-stack-create-events-parties-reference-picklist.json
This set of procedures does the following:
1. Preliminary JSON script
    b. N pre-NA-B rename BriefName column
    This script re-names column: "NamBriefName" to "workingName"
2. Apply the following JSON scripts:
    a. NA cleaning names – general
    b. NB cleaning names – specific
    c. N-XL lookup names (uses 'Event parties changelist xx')
    d. N-XD name spelling and clusters
    e. N-XSI strip initials
3. Follow up JSON scripts:
    c. NR remove columns – gets rid of scaffolding

This script removes the following column:
NameFullNamePreSort

d. N-post-NR-B replace BriefName
Return column names: “workingName” to “NamBriefName“
and ““strippedWorkingName” back to “strippedName”

**Export** as Excel file and save as 'EVENTS-ento methods parties partX xx'. This is used as input to the SITES procedure.

**Sites**

*Partially taken from Chapter 5 Sites, above*
See the MSWord doc: ‘Regular expressions for sites’ for the detailed GREL commands

**Input file**: 'EVENTS-ento methods parties partX xx'
**Procedure**: z-stack-create-events-sites-reference-picklist.json (parts 1&2, one after the other)

This set of procedures does the following:

1. Apply the following JSON scripts:
   a. PA cleaning places – general
   b. PB cleaning places – specific
   c. optional P-XC presenting changes to events-sites (optional – this can be omitted)
   d. P-XL lookup places (uses ‘Event sites changelist xx’)
   e. PD spelling cluster-edits corrections
   f. PG geography
   g. PC presenting changes to events-sites (important – removes prepositions, adds country, township, etc)

   Then run a second pass on the locations to get a better match. This removes things like “, near Sydney” from “Como, near Sydney”.
   h. P-SS2 site synonymy
   i. PC presenting changes to events-sites (important – removes prepositions, adds country, township, etc)
   j. PCX+ after PC for EVENTS (puts info from NoNear columns into Loc column)

   Note: ‘PC’ is the presentation module which can be run after ‘PB’ or ‘PD’.

2. Follow up JSON scripts: (JT: dunno about this)
   e. PR remove columns – gets rid of scaffolding
   This script removes the following columns:
   strippedOriginalLocation
   LocPreciseLocationPreSort

Leave the column headings for location for now. They will be fixed in the Events – combine step below.

Don't export yet. Run the events ordered procedure. It should be OK it doesn't use a lookup.

This is an optional step but there are two good reasons to run here:

1. The non-preferred events are labelled and can be removed
2. columns are re-ordered

**Procedure**: z-stack-create-events-ordered-reference-picklist.json

This set of procedures does the following:

1. Apply the following JSON scripts:
   a. JSON X reconstruct (run on the second pass locations) and re-arrange column order for Excel operations below.
   b. JSON X-WO reconstruct without something (extra columns with a key component left out – ie either date, method, site, or party are left out)

**Export** as Excel file and save as 'EVENTS-ento events partA xx'.

**Split – now rejoin**

Rejoin the two spreadsheets.

Note: The rows that contain CECs will be removed in the next step: 'Remove CECs'. For ento this is about 15k events (May 2013).

Save rejoined spreadsheet as 'EVENTS-ento events partsAB xx'

This spreadsheet will be greater than 65,000 rows. Save as .xlsx file

To join two worksheets, create a new worksheet, then cut-paste part A, then cut-paste partB

**Remove CECs**

Sort on column **ColEventCode**

Cut, remove and paste into a new worksheet

Name the worksheet CECs. Work on this in the CEC section

**Remove non-preferred events**

In column: **preferredEvent** remove = 'N', 'bounds'

(Ento: about 10,000 records removed 11/2013)

Remove columns –

These columns have been created before they are wanted. Remove them now. They will be recreated when we re-run the procedure: 'z-stack-create-events-ordered-reference-picklist.json'

If JSON X-WO was run there will be a bunch of superfluous columns. Pull out columns B-H.

Remove these columns:

eventReconstructed
strippedEvent
strippedWOdate
strippedWOparty
strippedWOsite
strippedWOmethod
preferredEvent

**Split for Malacology**

Malacology has too many CECs (about 95% of events have a CEC)

Hold your nerve. Jump to Events – combine below, run the procedure, remove the non-preferred events and create a smaller dataset. Then come back to here to add in duplicates. Still doesn't work. The dataset is too big.

**Reworking locations with 'near' in them and other redundant descriptions**
eg 'Bondi, near Sydney' becomes 'Bondi'
In Excel, open 'EVENTS ento reference – dates+methods+parties+sites'
The column **alteredLocation** (about column T) contains descriptors for what has been changed in the location

| alteredLocation | example | note |
|---|---|---|
| removed near | ", near Coffs Harbour" | keep |
| removed Sydney | "Bondi, Sydney" | keep |
| removed Blue Mountains | "Katoomba, Blue Mountains" | keep |
| removed Township | "Brisbane" | keep |
| removed districts | x, Iluka district | keep |
| excluded LocDistrictCountyShire | Christmas Island | keep |
| excluded LocTownship | Orange | keep |
| removed N-S-E-W | X National Park, S of Dubbo | keep |
| removed other | Tamborine Mount | keep |
| removed island groups | x Island, Great Barrier Reef | keep |

The column **facetLocation** was used as scaffolding. Not required here.

| facetLocation | example | note |
|---|---|---|
| near | "Murray River, near Echuca" | don't use here |
| extensive feature | lake, river, road | don't use here |
| extensive – road | road, street, highway, track | don't use here |

Copy rows where **alteredLocation** (about column I) has an entry eg: "removed x" such as "removed near" "removed Sydney" "removed Blue Mountains" "removed Township" "removed other", and 'excluded …' into another worksheet. Rename that worksheet 'secondary'
Don't pay attention to column **facetLocation** with its "near", "extensive feature" or "extensive – road"

In this second worksheet

- In column **CountOfUse** (about column "I"), give these events a low priority. They are really only pointers for other events
    - Subtract 100: due complications when all other locations=0
      Copy column=CountOfUse. In copied column use formula =(Y2-100)
      Copy and paste special = values back onto original
      Delete copied column

*Note column positions have now been changed for the following:*

- To help,
    - Colour orange **strippedLoc-D** (~G), **strippedLoc-NoNear** (~U), **LocationNoNear** (~T),
    - Colour blue **strippedLocation** (~H), and **LocPreciseLocation** (~G)
- Delete contents of **strippedLocation** (about column H)
- Sort on **strippedLoc-D** (about column U)
- Copy and paste contents of **strippedLoc-D** (about column U) into **strippedLocation** (~H) (check length of column) (these are the excluded townships and district – visually check)
- Sort on **strippedLoc-NoNear** (about column W)

- Copy and paste contents of **strippedLoc-NoNear** (about column W) into **strippedLocation** (~H) (check length of column)
- Sort on **LocationNoNear** (about column V)
- Copy and paste contents of **LocationNoNear** (about column V) into **LocPreciseLocation** (about column G) (check length of column) There will be some blanks, don't copy them across.

In the primary worksheet

- Check for and remove rows with ColEventCodes (about column F) (used elsewhere). See section **Collection Event Codes** below.

** Doesn't work for Malacology – 95% of malacology collection events have a CEC

Back in the secondary worksheet

- Re-sequence by adding 1,000,000.
  i. duplicate column A -> column B
  ii. formula =(a2+1000000)
  iii. copy values back to column A
  iv. delete column B
- Renumber sequence beginning with 1,000,001 (there are about 10,000 ento events 30/6/13)
- Copy entire worksheet and paste onto the bottom of the first worksheet.

## Elevation

*(Copied directly from EVENTS in previous chapter).*
Elevations are stored in EMu as both metres and feet.
Label data comes with elevation as either metres or feet.
As a way of marrying the two, duplicate those records with elevation, and have two elevations for each site – one for metres and one for feet, though without any reference to feet or metres.

When checking a site on a label for a match in the picklist, an elevation will match one of either the metre value or the feet value, either is okay as they both have the same SITE IRN.

**In the compound worksheet**

Duplicate the column **Elevation_metres** and paste in the column before. Rename new column **Elevation_NoUnits.** Give it a colour, eg yellow.
Copy the worksheet and paste into a new worksheet. Label it **elevation.**
In the **elevation** worksheet

7. In **elevation** worksheet throw away everything except those with an elevation
   a. Sort on Elevation_NoUnits
   b. Delete all those with an Elevation=0, or silly
   c. Chop off all those <u>without</u> an Elevation
   d. Copy column Elevation_feet on top of column Elevation_NoUnits.
   e. Renumber the sequence by adding 3,000,000 …
      i. duplicate column A -> column B
      ii. formula =(a2+3000000)
      iii. copy values back to column A
      iv. delete column B
   f. Colour entire worksheet green. (About 7,000 records for EVENTS)

8. Copy this entire **elevation** worksheet and paste onto the end of the first worksheet.
    i. Ctrl+A doesn't copy to another worksheet so this is workaround
    ii. select header row
    iii. ctrl+shift+down-arrow
    iv. Copy-paste
9. Follow up: sort on Elevation_NoUnits and delete all those with an Elevation=0
10. We now have a worksheet with a column labelled **Elevation_NoUnits.** This column contains the elevation twice - in both metres and feet - for each SITE with an elevation.

**Save it**

We now have a longer list of events. Ready to run EVENTS…
Save A+B worksheet as '**EVENTS-ento events partsAB extended xx'** (the full worksheet is too big for OpenRefine)
This is used as input to the EVENTS procedure.

**Events – combine the above Date + Name + Site + Method and sort**

This procedure presents preferred EVENTS using a sort criterion defined in 'JSON X'
See the MSWord doc: 'Regular expressions for events' for the detailed GREL commands

**Input file**: 'EVENTS-ento events partsA+B extended xx'
**Procedure**: z-stack-create-events-ordered-reference-picklist.json
This set of procedures does the following:
Apply the following JSON scripts:

1. JSON X reconstruct (run on the second pass locations)
   and re-arrange column order for Excel operations below.
2. JSON X-WO reconstruct without something (extra columns with a key component left out – ie either date, method, site, or party are left out)

**Export** as Excel file and save as EVENTS_PL reference xx'.
Use Excel
Copy these columns to another worksheet:

sequence
event_irn
eventReconstructed
strippedEvent
strippedWOdate
strippedWOparty
strippedWOsite
strippedWOmethod
preferredEvent
…
CountOfUse

Rename the worksheet, 'EVENTS_PL reference'. Make it page 1.
In column: **preferredEvent** remove = 'N', 'bounds'
In column **eventReconstructed** order alphabetically.
Check that there are no events with location = 'z' or 'x'

**Re-import** page 1 of the spreadsheet as 'EVENTS_PL reference xx' into OpenRefine

'**EVENTS_PL reference xx**' is now the reference picklist for EVENTS.
Put the reference to this in JSON V-EL and JSON V-WOL, and rebuild 'check-against-events'

## CEC – Collection Event Codes

Some collection events come with Collection Event Codes which are unique for a particular event. They look something like [97-53], or [BRITTON 2007068].

Modern Collection Event Codes are a good way to reliably match an event. If a modern record has an event code, then that is all you need to match, and other event info is irrelevant. But older event codes are not peculiar to a unique event, and so when trying to match event codes, append a date to them. ie assume that date-plus-event code is unique.

### Site-numbers

Some events come with a site-number, eg 'Site 16', or 'Site A', these appear in the EMu summary data in the same form as a CEC, ie with a square bracket around them. When detecting and managing site-numbers, treat them as CECs. So, the checklist of CECs will hold site-numbers, BVP data will hold site-numbers as CECs, and BVP site-numbers will be checked against CEC checklist. One of the last tasks with BVP data will be to separate the site-numbers out of the CECs.

### Procedure for cleaning and preparing Collection Event Codes

In section **Sites** (in this chapter) above, those events with an event code were put into a separate worksheet called **Event-codes**. Open this worksheet.

1. Save as 'EVENTS_CEC reference xx'
2. Import 'EVENTS_CEC reference xx' into OpenRefine

**Input file**: "EVENTS_CEC reference xx"
**Procedure**: z-stack-create-CEC-picklist.json
This set of procedures does the following:

1. Apply the following JSON scripts:
   a. JSON X-CEC event code reference.json (de-duplicates + cleans)

**'EVENTS_CEC reference xx'** is now the reference list for CECs.
Put the reference to this in JSON V-CECL, and rebuild 'check-against-events'

## Construct BVP events data and compare against reference picklist

This procedure does two things:

1. creates EVENTS info from a file exported from the Biodiversity Volunteer Portal
2. compares that EVENTS data with the EVENTS reference picklist above

### Construct BVP events data with Excel

BVP data comes as a series of spreadsheets. The spreadsheets contain date, collector, site, and method information.

**Filename**: Example 'BVP treehoppers xx '

Carry out the cleaning processes for DATES, METHODS, PARTIES, SITES and as detailed in Chapters 2-5 above
Fix any obvious spelling and other errors – eg things in wrong columns

Save spreadsheet as:
'BVP treehoppers repaired - dates+methods+parties+sites extended xx' - where xx is a sequence marker

## Process to check against EVENTS picklist

An EVENT is made up of a DATE, PARTY, SITE and METHOD.
Each of those entities is checked independently against their own standard:

- DATES are given a syntax check
- PARTIES are allocated a parties IRN if they match the parties reference
- SITES are allocated a site IRN if they match the sites reference
- METHODS are checked against a picklist, and adjusted if necessary

After cleaning, checking and allocating IRNs, an event is assembled and a stripped version of the event is compared against the EVENTS reference above 'EVENTS_PL reference xx'.

**Initial input file**: 'BVP treehoppers repaired xx'. Create a project in OpenRefine
Then run through the four following procedures to generate a cleaned and referenced spreadsheet: 'BVP treehoppers repaired - dates+methods+parties+sites xx'

**Dates**
*See Chapter 2 Dates: Clean BVP dates and split into begin and end dates*

**Methods**
*See Chapter 3 Methods: Clean BVP methods data and compare against reference picklist*

**Parties**
*See Chapter 4 Parties: Clean BVP parties data and compare against reference picklist*

**Sites**
*See Chapter 5 Sites: Clean BVP sites data and compare against reference picklist*

**Collection Event Codes and other strays**
Other things are added in at combine time (next step). This includes stuff like ColEventCode – Collection Event Codes.

**Events – combine the above Date + Name + Site + Method**
See the MSWord doc: 'Regular expressions for events' for the detailed GREL commands

**Input file**: 'BVP treehoppers repaired - dates+methods+parties+sites xx'
**Procedure**: z-stack-check-against-events-picklist.json

This set of procedures does the following:

1. Apply the following JSON scripts:
    a. V-X construct events
    b. V-EL lookup event IRN (uses EVENTS_PL reference xx)
    c. V-2X muster secondary event IRNs
    d. V-DL lookup matching dates – for info only

e. V-UL lookup import list (uses 'BVP EMu upload list so far xx')

Don't bother saving this – it is for checking only

**Input file**: 'BVP treehoppers repaired - dates+methods+parties+sites xx'
**Procedure**: z-stack-check-against-events-for-EMu.json
Return to step 1 in OpenRefine

1. Do step 1 above
2. Remove unnecessary columns:
   V-R-EMu readjust columns for EMu,
   and re-order according to table below
   **Export** as Excel file and save as 'BVP daymoths done for EMu xx'

**Input file**: 'BVP treehoppers repaired - dates+methods+parties+sites xx'
**Procedure**: z-stack-check-against-events-for-ALA.json
Return to step 1 in OpenRefine

1. Do step 1 above
2. Remove unnecessary columns:
   V-R-ALA readjust columns for ALA sandbox,
   and re-order according to table below
   **Export** as Excel file and save as 'BVP daymoths done for ALA xx'
   Even better, export as CSV file, open in a text editor and paste into sandbox.

**Input file**: 'BVP treehoppers repaired - dates+methods+parties+sites xx'
**Procedure**: z-stack-check-against-events-for-analysis.json
this procedure is detailed in the next section

Use Excel
Remove rows already uploaded in column **uploaded**.

'**BVP treehoppers done for EMu xx'** is the file that can be imported into EMu.
'**BVP treehoppers done for ALA xx'** is the file that can be imported into ALA sandbox.
'**BVP treehoppers done for analysis xx'** is the file that can be used to check for outliers. See next section.

## How many dates correspond to an event date?

Optional procedure to check the number of dates that exist as an event. This can give an indication of the potential number of matching events. This procedure can be included in the above set of procedure if useful.

**Input file**: 'BVP treehoppers repaired - dates+methods+parties+sites+events xx'
**Procedure**: JSON V-DL lookup matching event dates.json
This procedure creates an extra column with an 'x' where the date matches an event date.

1. Apply the following JSON scripts:
   a. V-DL lookup matching event dates

Remove the column 'matchingEventDate' when no longer required

## Result

The resulting BVP data file 'BVP treehoppers repaired – dates+methods+parties+sites+events done xx' now contains these columns:

| column | comment |
|---|---|
| | **Dates** |
| eventDateBegin | YYYY-MM-DD, or YYYY-MM-, or YYYY |
| eventDateEnd | YYYY-MM-DD, or YYYY-MM-, or YYYY |
| | |
| | **Parties** |
| identifiedBy_irn | EMu party IRN |
| recordedBy_irn | EMu party IRN |
| recordedByBriefName | recordedBy written briefly |
| strippedName | fingerprint of brief name of recordedBy – used for comparing against reference parties |
| collector1 | collector, or first person of a group |
| collector1_irn | EMu party IRN |
| collector2 | second collector of a group |
| collector2_irn | EMu party IRN |
| collector3 | third collector of a group |
| collector3_irn | EMu party IRN |
| collector4 | fourth collector of a group |
| collector4_irn | EMu party IRN |
| | **Sites** |
| site_irn | EMu site IRN |
| LocOcean | |
| LocCountry | |
| LocProvinceStateTerritory | |
| LocDistrictCountyShire | Islands as used by EMu |
| LocTownship | Township as used by EMu |
| strippedLocation | fingerprint representation of location – used for comparing against reference sites |
| LocPreciseLocation | Cleaned up version of verbatimLocality |
| | |
| | **Methods** |
| samplingProtocol-original | Methods from BVP |
| ColCollectionMethod | Cleaned up and checked methods |
| | |
| | **Events** |
| event_irn | EMu event IRN |
| eventReconstructed | Event compiled from cleaned date, party, site and method |
| strippedEvent | fingerprint of the reconstructed event – used for comparing against reference events |
| ColEventCode | Collection event code manually separated from event |
| | |
| | **Admin** |
| uploaded | This record has previously been imported |

… plus a bunch more as 'scaffolding' – used to help build and analyse the process.

## EMu requires these columns

| Column | Heading | comment |
|---|---|---|
| 0 | sequence | |
| 1 | occurrenceID | http + taskID |
| 2 | catalogNumber | K.12345 |
| 3 | cat_irn | Get Mikey to add IRNs to a list of K numbers |
| 4 | occurrenceRemarks | |
| 5 | scientificName | |
| 6 | originalNameUsage | |
| 7 | originalNameAuthorship | |
| 8 | dateIdentified | |
| 9 | identifiedBy | |
| 10 | identifiedBy_irn | EMu party IRN |
| 11 | typeStatus | |
| 12 | sex | gender – male, female |
| 13 | fieldNotes | eg in cop. |
| 14 | fieldNumber | eg #234, HH59 but not site numbers |
| 15 | transcriberNotes | |
| 16 | validatorNotes | |
| 17 | collectionCode | eg Greg Daniels Collection |
| 18 | event_irn | EMu event IRN as matched |
| 19 | eventDateBegin | YYYY-MM-DD, or YYYY-MM-, or YYYY |
| 20 | eventDateEnd | YYYY-MM-DD, or YYYY-MM-, or YYYY |
| 21 | eventReconstructed | Event compiled from cleaned date, party, site and method |
| 22 | ColCollectionMethod | Cleaned up and checked methods |
| 23 | ColEventCode | Collection event code manually separated from event |
| 24 | collector1 | collector, or first person of a group |
| 25 | collector1_irn | EMu party IRN |
| 26 | collector2 | second collector of a group |
| 27 | collector2_irn | EMu party IRN |
| 28 | collector3 | third collector of a group |
| 29 | collector3_irn | EMu party IRN |
| 30 | collector4 | fourth collector of a group |
| 31 | collector4_irn | EMu party IRN |
| 32 | site_irn | EMu site IRN as matched |
| 33 | LocOcean | |
| 34 | LocCountry | |
| 35 | LocProvinceStateTerritory | |
| 36 | LocDistrictCountyShire | Islands as used by EMu |
| 37 | LocTownship | Township as used by EMu |
| 38 | LocPreciseLocation | Cleaned up version of verbatimLocality |
| 39 | coordinateUncertaintyInMeters | |
| 40 | decimalLatitude | |
| 41 | decimalLongitude | |
| 42 | verbatimElevation | |
| 43 | verbatimElevationTo | |

| | | |
|---|---|---|
| 44 | verbatimLatitude | |
| 45 | verbatimLongitude | |
| 46 | minimumDepthInMeters | |
| 47 | maximumDepthInMeters | |
| 48 | habitat | |
| 49 | uploaded | This record has previously been imported<br>Remove after fixing |

## ALA sandbox uses these columns

These are the columns that ALA sandbox can use. Create using 'JSON V-R-ALA readjust columns for sandbox.json' starting after 'JSON V-UL lookup import list'

| Column | from OpenRefine | for ALA sandbox | comment |
|---|---|---|---|
| 0 | catalogNumber | catalogNumber | |
| 1 | taskID | occurrenceID | add http: |
| 2 | institutionCode | institutionCode | |
| 3 | basisOfRecord | basisOfRecord | |
| 4 | occurrenceRemarks | occurrenceRemarks | |
| 5 | scientificName | scientificName | |
| 6 | transcriberID | georeferencedBy | |
| 7 | transcriberNotes | georeferenceRemarks | |
| 8 | validatorNotes | validatorNotes | |
| 9 | collectionCode | collectionCode | |
| 10 | eventDate | eventDate | |
| 11 | eventReconstructed | eventRemarks | |
| 12 | ColCollectionMethod | samplingProtocol | |
| 13 | ColEventCode | fieldNumber | see note below |
| 14 | recordedBy | recordedBy | |
| 15 | LocOcean | waterBody | |
| 16 | LocCountry | country | |
| 17 | LocProvinceStateTerritory | stateProvince | |
| 18 | LocDistrictCountyShire | county | |
| 19 | LocTownship | municipality | |
| 20 | LocPreciseLocation | locality | |
| 21 | coordinateUncertaintyInMeters | coordinateUncertaintyInMeters | |
| 22 | decimalLatitude | decimalLatitude | |
| 23 | decimalLongitude | decimalLongitude | |
| 24 | verbatimElevation | verbatimElevation | no ElevationTo |
| 25 | verbatimLatitude | verbatimLatitude | |
| 26 | verbatimLongitude | verbatimLongitude | |
| 27 | minimumDepthInMeters | minimumDepthInMeters | |
| 28 | maximumDepthInMeters | maximumDepthInMeters | |
| 29 | habitat | habitat | |
| 30 | typeStatus | typeStatus | |
| 31 | sex | sex | |
| 32 | fieldNumber | fieldNotes | see note below |
| 33 | uploaded | uploaded | batch # |
| 34 | matchingDate | matchingDate | x |
| 35 | matchingCollectors | matchingCollectors | 1 2 3 4 |
| 36 | matchingSite | matchingSite | 'Site' |
| 37 | matchingEvent | matchingEvent | 'Event' |
| | | | |
| | **Probably not needed** | | |
| | event_irn | eventID | |
| | site_irn | locationID | |

**Note**: Notes about the event have been shuffled.

- ColEventCode is now called fieldNumber
- fieldNumber is now combined with fieldNotes
- verbatimElevationTo not converted to maximumElevationInMeters (units may be ft)

## Process to create a tool for manual checking

This process creates a method for a human to quickly add a relevant event IRN to a record that is a likely candidate.

The result of all the above semi-automated processes is a spreadsheet with records matched to IRNs where possible. This is an incomplete process, as many potential candidates are missed. A human eye can quickly pick up other similarities and outliers.

## Create a series of sub-standard matches to EVENTS

Match everything except for one criterion. ie we have almost got a match. Something may be spelled incorrectly or missing.

In order to match an EVENT, these four criteria need to be matched:

- DATE
- PARTY
- SITE
- METHOD

Part 1. Create a set of columns where records match three of the above four criteria.

| Column 1 | Column 2 | comment |
|---|---|---|
| strippedWOdate | woDate_irn | matches PARTY SITE METHOD |
| strippedWOparty | woParty_irn | matches DATE SITE METHOD |
| strippedWOsite | woSite_irn | matches DATE PARTY METHOD |
| strippedWOmethod | woMethod_irn | matches DATE PARTY SITE |

## Mark those records which should be investigated

Two parts. Part A, where dates are trusted and Part B, where dates are suspect.

## Part A. Remove records where dates are trusted

| Column | include / exclude | comment |
|---|---|---|
| event_irn | exclude | previously successful |
| uploaded | exclude | already done |
| matchingDate | include | an EVENT with that DATE exists (including EVENTS without DATES) |

**Input file**: 'BVP treehoppers repaired - dates+methods+parties+sites xx'
**Procedure**: z-stack-check-against-events-for-analysis.json
Return to step 1 in OpenRefine

1. Do step 1 as detailed in *Process to check against EVENTS picklist* above
1. Create columns to match 3 of 4 criteria
    a. V-WO reconstruct without something
    b. V-WOL lookup substandard event IRN
2. Mark records to be investigated
    a. V-RE readjust columns for Events spreadsheet

**Export** as Excel file and save as 'BVP cicadas done for analysis xx'

Use Excel

a. Rename primary worksheet as “BVP full”
b. Copy first 24 columns up to and including **ULmanualCheck** and paste into new worksheet. Name worksheet as “BVP selected columns”

These columns line up with columns of ‘EVENTS_PL reference xx’

| **Column** | **comment** |
|---|---|
| sequence | |
| manual_irn | column is ready to add |
| evetnReconstructed | |
| strippedEvent | |
| strippedWOdate | |
| strippedWOparty | |
| strippedWOsite | |
| strippedWOmethod | |
| occurrenceID | link to BVP record online at ALA |
| event_irn | |
| woDate_irn | |
| woParty_irn | |
| woSite_irn | |
| woMethod_irn | |
| matchingDate | includes dateless |
| uploaded | eg cicadas batch 1 21Feb13 |
| manualCheck | ‘Check party’, ‘Check site’, ‘Check method’ |

The following columns are duplicated and renamed to line up with the columns in ‘V-UL lookup import list’. They are tacked on as columns after the above set of columns.

| **Column** | **comment** |
|---|---|
| ULreg_no | Copy of catalogNumber |
| ULbatch | Copy of uploaded – needs to be changed below |
| ULevent_irn | Copy of event_irn |
| ULsequence | Copy of sequence |
| ULtaxon | Copy of scientificName |
| ULcollectionEvent | Copy of eventReconstructed |
| ULmanualCheck | Copy of manualCheck |

The remaining columns are useful as checks

**c.** Sort on column: **manualCheck**
d. Copy all rows with a **manualCheck** entry. Paste into a new worksheet. Name worksheet ‘BVP manual check’

Colour the 4 categories of manualCheck. Choose these colours..

| **manualCheck** | **colour** | **comment** |
|---|---|---|
| Check date | orange | Same everything except date |
| Check party | yellow | Someone else at the same place |
| Check site | green | Same person at a different place |
| Check method | blue | Same person and place, different method |

e. Open spreadsheet of EVENTS: ‘EVENTS_PL reference xx’

f. Save as 'muckup EVENTS_PL reference xx'. This is a scratch file, replace it when reference file is replaced.
g. Cut and paste all rows of interest (you don't need to use them all, eg just rows with dates is OK) from "BVP manual check" above, into bottom of 'muckup EVENTS_PL reference xx'. Confirm columns line up. Fill empty reference column headings with corresponding ones from BVP.
h. Sort on column: **eventReconstructed**

Prep work is done. Spreadsheet is loaded.

**Manual check procedure**
Use Excel spreadsheet 'muckup EVENTS_PL reference xx' loaded with selected BVP rows
Repeat this process

1. In column: **event_irn**
   Find gaps. Use CTRL-downarrow
2. In column: **eventReconstructed**
   Scan other rows with similar dates and look for a match in this column
3. When events match
   Copy irn from matching event into gap

Finished checking

**Extract all the working records from the reference**
Sort on column **CountOfUse**. This should send coloured records to the bottom of 'muckup EVENTS'
Cut off coloured records and paste into 'BVP cicadas done for analysis'. New worksheet
Name worksheet 'BVP tested'

We now have a full worksheet 'BVP full with tested' that has gone through a manual check and has some items changed.

## Option 1. Add manual IRNs to dataset for EMu

Add column **manual_irn** to original spreadsheets – baseline, or parties, sites or even events. Put it after **sequence**.

## Option 2. Add records to 'EMu upload list so far'

In spreadsheet: 'BVP cicadas done for analysis'
In worksheet: 'BVP tested'

1. Sort on column: manual_irn
2. Sort on column: uploaded
   This should reveal any manual IRNs which have not been uploaded.
3. Copy these rows (of new manual IRNs) and paste into a new worksheet in spreadsheet: 'EMu upload list so far'. Rename worksheet 'Cicadas manual check xx'

In spreadsheet 'EMu upload list so far'

4. In worksheet 'Cicadas manual check xx'
5. Copy column manual_irn into ULevent_irn (about column 'T')
6. Delete all columns up to ULreg_no (about column 'R')
7. Add an entry to ULbatch, eg 'Date check xx'
8. Copy the following seven columns and paste into worksheet 'Total uploads'

Spreadsheet= 'BVP Emu upload list so far xx' worksheet= 'Total uploads'

| Total uploads | From analysis | comment |
|---|---|---|
| Reg_No | ULreg_no | |
| Batch | ULbatch | Add this now |
| event_irn | ULevent_irn | New info from testing above |
| sequence | ULsequence | |
| Taxon | ULtaxon | |
| Collection Event | ULcollectionEvent | |
| manualCheck | ULmanualCheck | |

## Feed the manual fixes back into OpenRefine

1. Save spreadsheet 'Emu upload list so far xx'
2. Upload into OpenRefine
3. Change reference to 'Emu upload list so far xx' in 'JSON V-UL lookup upload list'
4. Re-run 'z-rebuild stack to check against events picklist'
5. Re-apply in OpenRefine 'z-stack-check-against-events-picklist'

## Part B. Untrusted dates

But what if the date is wrong…?

The procedure 'V-RE readjust columns for Events spreadsheet' has marked those records where the date is suspect, ie if you remove the DATE, then SITE, PARTY, and METHOD line up.

**Back in OpenRefine**
Run **Procedure**: z-stack-check-against-events-of-sub-standard.json
(It may have been run above)
In column **manualCheck** suspicious candidates for incorrect dates are marked 'Check date'
There may be overlap with 'Check method', 'Check party' and 'Check site', but those suspects are also marked.

Set up a filter:
In column **uploaded** filter on BLANK=TRUE (exclude those that are gone to EMu)
In column **woDate_irn** filter on BLANK=FALSE (everything but the date is OK)

In column **strippedWOdate**, Text facet
In column **strippedWOdate**

a. Work through the choices and check the dates as appearing in the images are as recorded. Use the hyperlinks to images on BVP.
b. Fix any records, and if necessary re-run all the processes

**Export** as Excel file and save as 'BVP cicadas done for analysis xx' (the same filename as for trusted dates above)
Use Excel

a. Rename primary worksheet as "BVP full"

b. Copy first 24 columns up to and including **ULmanualCheck** and paste into new worksheet. Name worksheet as “BVP selected columns”. (see above for column names)
c. …

Up to and including

d. Sort on column: **eventReconstructed**

Then, for dates, this is the line that is different

e. Sort on column: **strippedWOdate**

Prep work is done. Spreadsheet is loaded.

**Manual check procedure**

Same as above.