Biodiversity Volunteer Portal
Preparing BVP export data for import into EMu

OpenRefine procedures to clean data for EVENTS

An EVENT is made of DATE + PARTY + SITE + METHOD

Procedures for working with OpenRefine. These procedures are used to clean data about dates, collectors, locations and collection methods from the Biodiversity Volunteer Portal and make it compatible with data in the EMu EVENTS module.

Contents

D = Fix up and split the DATES

Uses file ‘BVP treehoppers original.csv’

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

In column: **eventDate**
Separate begin and end date
Create a new column called “eventDateEnd” based on “eventDate”
split(value, "/")[1]
Create a new column called “eventDateBegin” based on “eventDate”
split(value, "/")[0]

Fix up some errors, like 1928-3-11
in column: **eventDateBegin**
toString(toDate(value, 'yyyy-MM-dd'), 'yyyy-MM-dd')
if(length(value) <8, toString(toDate(value, 'yyyy-MM'), 'yyyy-MM-'), value)
add a hyphen

Same for column: **eventDateEnd**
toString(toDate(value, 'yyyy-MM-dd'), 'yyyy-MM-dd')
if(length(value) <8, toString(toDate(value, 'yyyy-MM'), 'yyyy-MM-'), value)

D-XL = lookup DATES

This procedure is used with the file 'EVENTS ento 51k'

Check against a list of dates where problems have been manually fixed
Before now:

- Create a spreadsheet called from the dates list – perhaps as a consequence of refining. Call the spreadsheet: "Event dates changelist xx"
  The spreadsheet has at least two columns:
  - one column: "bad" with the existing spelling
  - another column: "good" with the preferred spelling

- Create a OpenRefine project called: "Event dates changelist xx".
  where xxx is the latest version. Note: do not repeat version names.

Back to the project... 'EVENTS ento 51k xx'
In column: **ColDateVisitedFrom**

cell.cross("Event dates changelist xx", "bad").cells["good"].value[0]
cell.cross("Event dates changelist 2801a", "bad").cells["good"].value[0]

In column: **ColDateVisitedTo**

cell.cross("Event dates changelist xx", "bad").cells["good"].value[0]
cell.cross("Event dates changelist 2801a", "bad").cells["good"].value[0]

V-DL = lookup matching EVENT dates

This procedure is used with the file 'BVP treehoppers repaired – stuff xx'

Check against a list of EVENTS dates

Before now:

- Create a spreadsheet from the events list. Call the spreadsheet: "EVENTS-ento reference - dates". To keep the list not-too-long, make a list of unique dates. The spreadsheet has a column:
    - one column: "ColDateVisitedFrom"

- Use the OpenRefine project called: "EVENTS-ento reference – dates xx" where xx is the latest version. Note: do not repeat version names.

Back to the project... 'BVP treehoppers repaired – xx'
Create a new column called **matchingDate** based on **eventDateBegin**
(cell.cross("EVENTS mal reference dates 0306a",
"ColDateVisitedFrom").cells["ColDateVisitedFrom"].value[0])

Replace the date with an x
In column **matchingDate**
if(isBlank(value), NULL, "x")

Remove the column after counting

M = Fix up METHODS

Used in both the EVENTS file 'EVENTS ento 51k' used to create a picklist, and the BVP file 'BVP treehoppers xx'
The JSON for this appears in the script: JSON M methods

Use column: **ColCollectionMethod**

Change first letter to Uppercase
replace(value, /^[a-z]/, toUppercase(value[0]))

Get rid of quote symbols
replace(value, /[\"\']/, "")

Fix-up specific
A few strays:
replace(value, /[Mm][\.]?[\s]?[Vv][\.]?/, "MV")
replace(value, /^.*[Ss](weep).*$/, "Sweeping")
replace(value, /^.*[Ss](wept).*$/, "Sweeping")
replace(value, /^.*(MV light).*$/, "MV lamp")
replace(value, /^.*(MV Light).*$/, "MV lamp")
replace(value, /^.*(MV Lamp).*$/, "MV lamp")
replace(value, /^.*(bred).*$/i, "Reared")
replace(value, /^.*(larva).*$/i, "Reared")
replace(value, /^.*(pupa).*$/i, "Reared")
replace(value, /^.*(hatch).*$/i, "Reared")
replace(value, /^.*[Mm](alaise).*$/, "Malaise trap")
replace(value, /^.*[Yy](ellow).*(pan).*$/, "Yellow pan")
replace(value, /^.*[Ss](ticky).*(trap).*$/, "Sticky trap")
replace(value, /^.*(pit).*(trap).*$/i, "Pitfall trap")
replace(value, /^.*(fruit).*(trap).*$/i, "Fruit trap")
replace(value, /^.*(banana).*(trap).*$/i, "Fruit trap")

If the word 'or' doesn't exist then (Don't replace a legitimate phrase)
replace(value, /^.*(found|dead).*$/i, "Found or dead")

Use OpenRefine to clean this list up.

Brute approach – minimise as much as you can.
replace(value, /^.*[Tt][Rr][Aa][Ww][Ll].*$/, "Trawl")
replace(value, /^.*[Ss](eine).*$/, "Seine net")
replace(value, /^.*(Rotenone).*$/, "Rotenone")
replace(value, /^.*[Aa](irlift).*$/, "Airlift")
replace(value, /^.*[Dd](redge).*$/, "Dredged")
replace(value, /^.*\b[Ll](ine).*$/, "Hook and line")
replace(value, /^.*[Pp](oison).*$/, "Poison")
replace(value, /^.*[Ss](ieve).*$/, "Sieve")

replace(value, /^.*[Nn][Ee][Tt].*$/, "Net")

```
replace(value, /^.*[Ss](crap)[ei].*$/, "Scraped")
replace(value, /^.*[Ss](coop)[ei]?.*$/, "Scoop")
replace(value, /^.*[Gg](rab).*$/, "Grab")
replace(value, /^.*[Ww](ash).*$/, "Washing")
replace(value, /^.*[Cc](or)[ie].*$/, "Corer")
replace(value, /^.*[Pp](ick).*$/, "By hand")
replace(value, /^.*[Jj](ig).*$/, "Jigged")
replace(value, /^.*[Ll](ight).*$/, "At light")
replace(value, /^.*[Ff](ishing).*$/, "Hook and line")
replace(value, /^.*[Bb](each).*$/, "Found or dead")
replace(value, /^.*(SCUBA).*$/, "On SCUBA")
replace(value, /^.*[Ss](norkel).*$/, "On snorkel")
replace(value, /^.*[Hh](and).*$/, "By hand")
replace(value, /^.*[Ss](led).*$/, "Sledge")
replace(value, /^.*[Nn](ight).*$/, "At night")
```

M-XL = lookup METHODS

Used in both the EVENTS file ‘EVENTS ento 51k’ used to create a picklist, and the BVP file ‘BVP treehoppers xx’
The JSON for this appears in the script: JSON M-XL lookup methods

Check against a list of places where problems have been manually fixed
Before now:

- Create a spreadsheet from the places list – perhaps as a consequence of refining.
  Call the spreadsheet: “Event methods changelist xx”
  The spreadsheet has at least two columns:
  - one column: “bad” with the existing spelling
  - another column: “good” with the preferred spelling

- Create a OpenRefine project called: “Event methods changelist xx”.
  where xx is the latest version. Note: do not repeat version names.

In column: **ColCollectionMethod**
cell.cross("Event methods changelist xx", "bad").cells["good"].value[0]
eg:
cell.cross("Event methods changelist 2905a", "bad").cells["good"].value[0]

V-ML = lookup METHOD picklist

The JSON for this appears in the script 'JSON V-ML lookup method picklist' and expects 'JSON M methods' to have been run after initial preparation of the BVP file.

This procedure compares the methods recorded for an event against a picklist of known methods.

Methods will require some human intervention, as there are no good standard ways of referring to them.

Preparation
In Excel: Create a new column called "**ColCollectionMethod"** based on "**samplingProtocol"**
Run "JSON M" on this spreadsheet.

This procedure
In column: ColCollectionMethod
Filter on only those records with a collection method:

- Facet / Customized facets / Facet by blank
- Facet on blank=FALSE

Check against reference created above
if(isBlank(cell.cross("METHODS_PL reference 0905a1", "Method").cells["Method"].value[0]), "x", value)
This puts an "x" against unrecognised methods.

Manual step
Either

- export this file and change if required
- re-adjust the changelist "Event methods changelist xx". Just add a good-bad pair to the end of the file.
- Edit/delete directly in OpenRefine (select 'x', edit, delete the 'x')

M-SS = strip and synonymise methods

Used with METHODS procedure

Create a new column called "**strippedMethod"** based on "**ColCollectionMethod"**
value

Synonymies
At light = UV light = MV lamp
replace(value, /(MV lamp)|(At light)|(UV light$)/, "light")
fingerprint(value)

V-UL = lookup upload list

A spreadsheet is maintained that monitors all the records that have been uploaded into EMu.

Records are uploaded in batches
For example, a batch of 3000+ events was uploaded to EMu in 2012. They are listed in the spreadsheet 'Digivol Import 1 summary for John 2801a'

In general, each batch that is uploaded will have its own uniqueness.

Spreadsheet columns

Header	example	comment
Reg_No	K.12345	OPENREFINE looks for this
Batch	Digivol 1 2012	change this for each batch
event_irn	123456	
Taxon	eg Hemiptera	
Collection Event	summary	

Create a new column called **uploaded** based on **catalogNumber**

cell.cross("Digivol Import 1 summary 2801a", "Reg_No").cells["Reg_No"].value[0]
cell.cross("BVP EMu upload list so far 0403a", "Reg_No").cells["Batch"].value[0]

Create a new column called **manual_irn** based on **catalogNumber**
This contains every IRN that we know of so far – not just the manual ones.
cell.cross("BVP EMu upload list so far 0403a", "Reg_No").cells["event_irn"].value[0]

Not needed…
In column: **uploaded** facet on BLANK=FALSE (either uploaded or ready to go)
In column: **event_irn** facet on BLANK=TRUE (no event IRN)
In column: **event_irn**
cell.cross("BVP EMu uploads so far 0403a",
cells["Reg_No"].value).cells["event_irn"].value[0]

X = Reconstruct a cleaned event

In the reference file: 'EVENTS ento 51k'

This procedure is similar to the process 'V-X' below, with some column names changed. Unlike 'V-X', this procedure then sorts so as to get rid of duplicates, and choose preferred events.

This procedure assumes that the reference file has already had the dates, names, places and methods cleaned, and a stripped version of names and places created.

Remove columns **eventReconstructed** and **strippedEvent**. Removing these columns allows this procedure to be run again.

Create a new column called "**eventReconstructed**" based on "**event_irn**"
"" (dummy)

In column: **eventReconstructed**
value + cells["ColDateVisitedFrom"].value
value + ", " + cells["NamBriefName"].value
value + ", " + cells["LocPreciseLocation"].value (didn't work, but works now)
didn't work: for column **eventReconstructed** if(isNonBlank(value), value, "")
if(isBlank(cells["LocPreciseLocation"].value), value, value + ", " +
cells["LocPreciseLocation"].value)
value + ", " + cells["LocTownship"].value
value + ", " + cells["LocDistrictCountyShire"].value
value + ", " + cells["LocProvinceStateTerritory"].value
value + ", " + cells["LocCountry"].value
value + ", " + cells["LocOcean"].value
value + " " + cells["Elevation_NoUnits"].value
value + ", " + cells["ColCollectionMethod"].value
value + ", " + cells["ColEventCode"].value

get rid of double commas, and leading commas
replace(value, " , ", "")
replace(value, ",,", ",") (repeat)
replace(value, /^(,)/, "")
replace(value, "| ", "and")

Create a new column called "**strippedEvent**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedEvent**
value + cells["ColDateVisitedFrom"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedLocation"].value
value + " " + fingerprint(cells["strippedMethod"].value)

trim and collapse consecutive whitespace

Secondary priority for sorting

so that this is repeatable, sort initially on IRN

sort on: collevent _irn (numbers – lowest first) (one for events)

order alphabetically – all cleaned locations next = discarded locations will be in order

sort on: LocPreciseLocationPreSort (alphabetical sort)

order alphabetically – the clean locations that will be used – shove the empties off

sort on: LocPreciseLocation (alphabetical sort)

order by person

sort on: NamBriefName (alphabetical sort)

order by date

sort on: ColDateVisitedFrom

order by event

sort on: eventReconstructed

Primary priority for sorting

sort on: CountOfObjects (numbers, largest first) - uppercase

sort on: CountofObjects (numbers, largest first) – lowercase

sort on: strippedOriginalLocation (alphabetic sort, BLANKS FIRST to give the blank Townships a high priority – there are a lot of them.)

ie Dubbo=LocTownship has a higher priority than Dubbo=LocPreciseLocation

sort on: strippedEvent

blank down on: strippedEvent (clear now, fill later)

Mark the preferred events

create a new column called "**preferredEvent**" based on "**strippedEvent**"

if(isBlank(value), "N", "Y") (set ON ERROR = blank)

"Y" to keep, "N" to remove

repeat this for when the column **preferredEvent** already exists:

In column: **preferredEvent**

if(isBlank(cells["strippedEvent"].value), "N", value)

In column: **preferredEvent**

if(cells["OutOfBounds"].value == "bounds", "bounds", value) (set ON ERROR = keep original)

In column: **preferredEvent** if there is a dodgy location eg '?', then not a preferred event

if(cells["LocPreciseLocation"].value == "x", "N", value) (set ON ERROR = keep original)

if(cells["LocPreciseLocation"].value == "z", "N", value) (set ON ERROR = keep original)

Re-order columns to make Excel operations easier

Use Excel to remove columns if needed

Call this file "**EVENTS_PL reference xx**"

X-WO = reconstruct an EVENT and leave something out

This is a similar procedure to V-WO, below. This is for the EVENT data.
The difference between V-WO and X-WO is the name of the data field: **eventDateBegin** v **ColDateVisitedFrom**.

Used to create an almost-event, ie one where a component has been left out. Use this as a sub-reference to test the BVP against.

Create a new column called "**strippedWOdate**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOdate**
value + " " + cells["strippedName"].value
value + " " + cells["strippedLocation"].value
value + " " + cells["strippedMethod"].value

trim and collapse consecutive whitespace

Create a new column called "**strippedWOparty**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOparty**
value + cells["ColDateVisitedFrom"].value
value + " " + cells["strippedLocation"].value
value + " " + cells["strippedMethod"].value

trim and collapse consecutive whitespace

Create a new column called "**strippedWOsite**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOsite**
value + cells["ColDateVisitedFrom"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedMethod"].value

trim and collapse consecutive whitespace

Create a new column called "**strippedWOmethod**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOmethod**
value + cells["ColDateVisitedFrom"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedLocation"].value

trim and collapse consecutive whitespace

V-WO = reconstruct a BVP EVENT and leave something out

This is a similar procedure to X-WO. This is for the BVP data.
The difference between V-WO and X-WO is the name of the data field: eventDateBegin v ColDateVisitedFrom.

Used to create an almost-event, ie one where a component has been left out. Use this as a sub-reference to test the BVP against.

Create a new column called "**strippedWOdate**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOdate**
value + " " + cells["strippedName"].value
value + " " + cells["strippedLocation"].value
value + " " + cells["strippedMethod"].value

trim and collapse consecutive whitespace

Create a new column called "**strippedWOparty**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOparty**
value + cells["eventDateBegin"].value
value + " " + cells["strippedLocation"].value
value + " " + cells["strippedMethod"].value

trim and collapse consecutive whitespace

Create a new column called "**strippedWOsite**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOsite**
value + cells["eventDateBegin"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedMethod"].value

trim and collapse consecutive whitespace

Create a new column called "**strippedWOmethod**" based on "**eventReconstructed**"
"" (dummy)

In column: **strippedWOmethod**
value + cells["eventDateBegin"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedLocation"].value

trim and collapse consecutive whitespace

V-X = construct an EVENT from date+party+site+method

For the BVP exported and cleaned file: 'BVP treehoppers repaired'

This procedure is similar to the process 'X' above, with some column names changed. This procedure does not do any sorting, as it does not need to get rid of duplicates.

Create a new column called "**eventReconstructed**" based on "**eventDateBegin**"
"" (dummy)

In column: **eventReconstructed**
value + cells["eventDateBegin"].value
value + ", " + cells["recordedBy"].value (recordedBy brief name)
value + ", " + cells["LocPreciseLocation"].value (didn't work, but OK now)
value + ", " + cells["LocTownship"].value
value + ", " + cells["LocDistrictCountyShire"].value
value + ", " + cells["LocProvinceStateTerritory"].value
value + ", " + cells["LocCountry"].value
value + ", " + cells["LocOcean"].value
value + " " + replace(cells["verbatimElevation"].value, /[\s].*$/, "") (elevation without units)
value + ", " + cells["ColCollectionMethod"].value
value + ", " + cells["ColEventCode"].value

get rid of double commas, and leading commas
replace(value, " ,", "") re-transform
replace(value, /^(,)/, "") re-transform

trim and collapse consecutive whitespace

Fingerprint events – using synonymous locations

SITES module created several synonymies of locations. Viz locations with district removed, and locations with 'near xx' removed. Use these different locations to create different events.
Note: Collection Event Codes are now done separately. See V-CEC

Create a new column called "**strippedEvent**" based on "**eventReconstructed**"
"" (dummy)
In column: **strippedEvent**
value + cells["eventDateBegin"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedLocation"].value
value + " " + cells["strippedMethod"].value
value + " " + fingerprint(cells["ColEventCode"].value) (no room for CEC)
trim and collapse consecutive whitespace

Create a new column called "**strippedEvent1**" based on "**strippedEvent**"
value (make it a copy – to be used as a reference later)

Create a new column called "**strippedEvent2**" based on "**strippedEvent1**"
"" (dummy)

In column: **strippedEvent2**
value + cells["eventDateBegin"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedLoc-D"].value
value + " " + cells["strippedMethod"].value
value + " " + fingerprint(cells["ColEventCode"].value) (no room for CEC)
trim and collapse consecutive whitespace

Create a new column called "**strippedEvent3**" based on "**strippedEvent2**"
"" (dummy)
In column: **strippedEvent3**
value + cells["eventDateBegin"].value
value + " " + cells["strippedName"].value
value + " " + cells["strippedLoc-NoNear"].value
value + " " + cells["strippedMethod"].value
value + " " + fingerprint(cells["strippedMethod"].value)
value + " " + fingerprint(cells["ColEventCode"].value) (no room for CEC)
trim and collapse consecutive whitespace

Note: Although reconstructing an event now includes a collection event code (ColEventCode), there is no corresponding ColEventCode in the reference. I think that these will be quite infrequent, and by rejecting them at this stage, forces them to be matched by hand.
Late note. Matching Collection Event Codes, now occurs. See V-CEC and V-CECL
Should I remove ColEventCode matching from this procedure?

V-EL = lookup EVENT irn

For the BVP exported and cleaned file: 'BVP treehoppers repaired'

This procedure follows 'V-X construct event' above and looks up the IRN of an event in a reference picklist previously created 'EVENTS_PL reference xx'.

Get rid of old event_irn so you can start afresh
Remove column: **event_irn**

Create a new column called "**event_irn**" based on "**strippedEvent**"
cell.cross("EVENTS_PL reference 3001a", "strippedEvent").cells["event_irn"].value[0]

… and I got 191 rows (21 Nov)

… and an understandably shorter 90 rows (23 Dec – problems with names reconcile)

… and then 41 on 27/12

… this Is a OPENREFINE problem. It shouldn't run a lookup more than once without restarting.

Create a new column called "**event_irn1**" based on "**event_irn**"
value
Create a new column called "**event_irn2**" based on "**strippedEvent2**"
cell.cross("EVENTS_PL reference 3001a", "strippedEvent").cells["event_irn"].value[0]
Create a new column called "**event_irn3**" based on "**strippedEvent3**"
cell.cross("EVENTS_PL reference 3001a", "strippedEvent").cells["event_irn"].value[0]

Note: If the reference picklist changes, its new name needs to be altered here (6x).

V-WOL = lookup substandard EVENT IRN

This procedure follows V-EL and then X-WO and tries to match with an event that is without a key component. Use this to see if the BVP data is missing a key component.

Create a new column called “**woDate_irn**” based on “**strippedWOdate**”
cell.cross("EVENTS_PL reference 2502a", "strippedWOdate").cells["event_irn"].value[0]

Create a new column called “**woParty_irn**” based on “**strippedWOparty**”
cell.cross("EVENTS_PL reference 2502a", "strippedWOparty").cells["event_irn"].value[0]

Create a new column called “**woSite_irn**” based on “**strippedWOsite**”
cell.cross("EVENTS_PL reference 2502a", "strippedWOsite").cells["event_irn"].value[0]

Create a new column called “**woMethod_irn**” based on “**strippedWOmethod**”
cell.cross("EVENTS_PL reference 2502a", "strippedWOmethod").cells["event_irn"].value[0]

V-2X = muster secondary event IRNs

Run this after running 'V-EL lookup event IRN' which finds more events by including variations of location.

Make a master list of site IRNs

In column: **event_irn**
Filter on empty cells: Facets / Customized facets / Facet by blank / BLANK = TRUE

cells["event_irn2"].value (add in the unique IRNs for events without Districts)
cells["event_irn3"].value (add in the unique IRNs for events without 'near')
(Note: manual irns may be selected after this process has run. This can be run multiply)
cells["manual_irn"].value (add in the unique IRNs for events selected manually)
cells["eventID"].value (add in the unique IRNs for events selected from BVP)

In column: **event_irn** remove filter (Collection Event Codes should over-ride others)
cells["CEC_irn"].value (add in the unique IRNs for events collection event code)

Column **event_irn** now contains the IRNs of matching events from EMu.
remove filter

Reporting

Add a series of columns that will inform completeness.

Event
Facet on **event_irn** BLANK = FALSE
Create a new column called **matchingEvent** based on **event_irn**
"Event"

Site
Facet on **site_irn** BLANK = FALSE
Create a new column called **matchingSite** based on **site_irn**
"Site"

Collectors
Facet on **recordedBy** BLANK = FALSE
Create a new column called **matchingCollectors** based on **recordedBy**
"x"

Facet on **collector1_irn** BLANK = FALSE
In column **matchingCollectors**
"1"
remove facets

Facet on **collector2** BLANK = FALSE
Facet on **collector2_irn** BLANK = FALSE
In column **matchingCollectors**
value + " 2"

Facet on **collector2_irn** BLANK = TRUE
In column **matchingCollectors**
value + " x"
remove facets

Facet on **collector3** BLANK = FALSE
Facet on **collector3_irn** BLANK = FALSE
In column **matchingCollectors**
value + " 3"
Facet on **collector3_irn** BLANK = TRUE
In column **matchingCollectors**
value + " x"
remove facets

Facet on **collector4** BLANK = FALSE
Facet on **collector4_irn** BLANK = FALSE
In column **matchingCollectors**
value + " 4"
Facet on **collector4_irn** BLANK = TRUE
In column **matchingCollectors**
value + " x"
remove facets

V-WEB = create a weblink to BVP

Adds a link that takes you directly to the web page on ALA for that task.
Run this at preparation time, creating a link to check any record quickly.

In column: taskID
"http://volunteer.ala.org.au/validate/task/" + value (this can be validated)
"http://volunteer.ala.org.au/task/show/" + value (this can be edited)
Rename column **taskID** to **occurrenceID**

Change taskID so that it takes you straight to a validation page on ALA
In column: **taskID**, transform to text (Edit cells / Common transforms / To text)
Create a new column called **occurrenceID** based on **taskID**
"http://volunteer.ala.org.au/task/show/" + value

Move column **taskID** to the end (keeps all the column numbers in line)

Create a hyperlink in Excel

Two ways

1. Create an extra column
2. Use a Macro

Hyperlink: Create an extra column

Add a column
Formula =HYPERLINK(A1)

Hyperlink: Use a macro

From: http://www.niallflynn.com/random-news/convert-urls-to-clickable-links-in-excel/

```
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’, right-click it, select ‘Insert’, then select ‘Module’.
- Paste the 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 :)

V-RE = readjust columns to compare against EVENTS picklist

Re-arrange columns to be in a position for analysis

Create a new column called **manual_irn** based on **sequence**
Move **eventReconstructed** to position 2
Move **strippedEvent** to position 3
Move **strippedWOdate** to next position
Move **strippedWOparty** to next position
Move **strippedWOmethod** to next position
Move **occurrenceID** to next position
Move **event_irn** to next position
Move **woDate_irn** to next position
Move **woParty_irn** to next position
Move **woSite_irn** to next position
Move **woMethod_irn** to next position
Move **matchingDate** to next position
Move **uploaded** to next position

Part a. Trusted dates

Create a new column called **manualCheck** based on **occurrenceID**

In column matchingDate add in the dateless
In column **eventDate** filter on BLANK=TRUE (dateless)
In column **matchingDate**
"dateless"
In column **eventDate** remove filter
(JT: 534)

For the next three steps:
In column **event_irn** filter on BLANK=TRUE (exclude those that are matched)
In column **uploaded** filter on BLANK=TRUE (exclude those that are gone to EMu)
In column **matchingDate** filter on BLANK=FALSE (a valid date is needed)

In column **woParty_irn** filter on BLANK=FALSE (everything but the party is OK)
In column **manualCheck**
"Check party"
In column **woParty_irn** reset filter
(JT:44)

In column **woSite_irn** filter on BLANK=FALSE (everything but the site is OK)
In column **manualCheck**
"Check site"
In column **woSite_irn** reset filter
(JT:351)

In column **woMethod_irn** filter on BLANK=FALSE (everything but the method is OK)
In column **manualCheck**
"Check method"
In column **woParty_irn** reset filter

(JT:12)

Part b. Untrusted dates

But what if the date is wrong…?

> Perhaps do dates after the others have been checked
> I think I need to do manual checks on all of the unlikely dates – ie
> Leave out **matchingDate** filter
> all those that match except for a date – whether or not there is a matching date or not

New filter set
For the next three steps:
In column **event_irn** filter on BLANK=TRUE (exclude those that are matched)
In column **uploaded** filter on BLANK=TRUE (exclude those that are gone to EMu)
In column **matchingDate** filter on BLANK=FALSE (a valid date is needed)
(Leave out dates completely – they may match the date of an existing event, but be wrong)

In column **woDate_irn** filter on BLANK=FALSE (everything but the date is OK)
In column **manualCheck**
if(isBlank(value), "Check date", value + " + Check date")
In column **woDate_irn** reset filter

Fix for today
replace(value, "validate/task/", "task/show/")

V-R-EMu = readjust columns for EMu

Prepare a set of about 40 columns, needed for import into EMu.
Change taskID so that it takes you straight to a the transcription page on ALA
In column: **taskID**
"http://volunteer.ala.org.au/task/show/" + value
Rename column **taskID** to **occurrenceID**

Mark duplicate SITES and EVENTS
Sites

In column **site_irn** Filter on BLANK=TRUE (only work with records without a site irn)
In column **strippedLocation** filter on DUPLICATES=TRUE
Create a new column called **duplicatedSite** based on **site_irn**
"Duplicated site"
Remove filters

Events

In column **event_irn** Filter on BLANK=TRUE (only work with records without an event irn)
In column **strippedEvent** filter on DUPLICATES=TRUE
Create a new column called **duplicatedEvent** based on **event_irn**
"Duplicated event"
Remove filters

Only one set of lat-long
For those records with a verbatim lat-long, remove the decimal lat-long
In column **verbatimLatitude** Filter on BLANK=FALSE
In column **decimalLatitude** delete values
In column **decimalLongitude** delete values

Remove columns
Remove column: **strippedEvent**
Remove column: **strippedName**
Remove column: **strippedLocation**
Remove column: **LocPreciseLocation**
Rename column: **LocOKPreciseLocation** to **LocPreciseLocation**

Don't know about these yet
Remove column: **eventReconstructed**
Rename column: **eventOKReconstructed** to **eventReconstructed**

EMu doesn't need these:
Remove column: **LocPreciseLocationPreSort**
Remove column: **transcriberID**
Remove column: **validatorID**
Remove column: **externalIdentifier**
Remove column: **exportComment**
Remove column: **basisOfRecord**
Remove column: **strippedOriginalLocation**
Remove column: **eventDate**

Remove column: **matchingDate**
Remove column: **eventMatch**
Remove column: **institutionCode**
Remove column: **locality**
Remove column: **recordedBy**
Remove column: **recordedBy_irn**
Remove column: **recordedByBriefName**
Remove column: **strippedOKLocation**
Remove column: **recordedByID**
Remove column: **verbatimLocalityID**
Remove column: **verbatimLocality-original**
Remove column: **samplingProtocol-original**

Reorder columns

These are the columns required by EMu:

Column	Heading	comment
0	**sequence**	
1	**taskID**	
2	**catalogNumber**	
3	**occurrenceRemarks**	
4	**scientificName**	
5	**originalNameUsage**	
6	**originalNameAuthorship**	
7	**dateIdentified**	
8	**identifiedBy**	
9	**identifiedBy_irn**	
10	**typeStatus**	
11	**transcriberNotes**	
12	**validatorNotes**	
13	**collectionCode**	
14	**event_irn**	
15	**eventDateBegin**	
16	**eventDateEnd**	
17	**eventReconstructed**	
18	**ColCollectionMethod**	
19	**ColEventCode**	
20	**collector1**	
21	**collector1_irn**	
22	**collector2**	
23	**collector2_irn**	
24	**collector3**	
25	**collector3_irn**	
26	**collector4**	
27	**collector4_irn**	
28	**site_irn**	
29	**LocOcean**	
30	**LocCountry**	
31	**LocProvinceStateTerritory**	
32	**LocDistrictCountyShire**	

33	**LocTownship**	
34	**LocPreciseLocation**	
35	**coordinateUncertaintyInMeters**	
36	**decimalLatitude**	
37	**decimalLongitude**	
38	**verbatimElevation**	
39	**verbatimLatitude**	
40	**verbatimLongitude**	

V-R-ALA = readjust columns for ALA sandbox

Done much earlier

Change taskID so that it takes you straight to a validation page on ALA
In column: taskID
"http://volunteer.ala.org.au/validate/task/" + value
Rename column **taskID** to **occurrenceID**

Shuffle notes about the event

- **fieldNumber** to be combined with **fieldNotes**
- **ColEventCode** to be called **fieldNumber**

In column **fieldNumber**
value + " " + cells["**fieldNotes**"].value
Remove column: **fieldNotes**
Rename column: **fieldNumber** to **fieldNotes**
Rename column: **ColEventCode** to **fieldNumber**

Remove columns
Remove column: **strippedEvent**
Remove column: **strippedName**
Remove column: **strippedLocation**
Remove column: **LocPreciseLocation**
Rename column: **LocOKPreciseLocation** to **LocPreciseLocation**

Don't know about these yet
Remove column: **eventReconstructed**
Rename column: **eventOKReconstructed** to **eventReconstructed**

This list is no longer up to date: See overview document
Remove column: **LocPreciseLocationPreSort**
Remove column: **transcriberID**
Remove column: **validatorID**
Remove column: **externalIdentifier**
Remove column: **exportComment**
Remove column: **strippedOriginalLocation**
Remove column: **eventDate** (I currently use eventDateBegin. eventDate might be better)
Remove column: **matchingDate**
Remove column: **eventMatch**
Remove column: **locality**
Remove column: **recordedBy_irn**
Remove column: **recordedBy**
Remove column: **strippedOKLocation**
Remove column: **recordedByID**
Remove column: **verbatimLocalityID**
Remove column: **verbatimLocality-original**
Remove column: **sequence**
Remove column: **taskID**
Remove column: **originalNameUsage**
Remove column: **collector1**

Remove column: **collector1_irn**
Remove column: **collector2**
Remove column: **collector2_irn**
Remove column: **collector3**
Remove column: **collector3_irn**
Remove column: **collector4**
Remove column: **collector4_irn**

Reorder columns

These are the columns that ALA sandbox can use
this list is not up-to-date. See overview document

Column	**from OpenRefine**	**for ALA sandbox**	
0	catalogNumber	catalogNumber	
1	institutionCode	institutionCode	
2	basisOfRecord	basisOfRecord	
3	occurrenceRemarks	occurrenceRemarks	
4	scientificName	scientificName	
5	dateIdentified	dateIdentified	
6	identifiedBy	identifiedBy	
7	typeStatus	typeStatus	
8	transcriberNotes	georeferenceRemarks	
9	validatorNotes	validatorNotes	
10	collectionCode	collectionCode	
11	event_irn	eventID	
12	eventDateBegin	eventDate	
13	eventDateEnd	eventDateEnd	
14	eventReconstructed	eventRemarks	
15	ColCollectionMethod	samplingProtocol	
16	ColEventCode	fieldNumber	
17	recordedBy	recordedBy	
18	site_irn	locationID	
19	LocOcean	waterBody	
20	LocCountry	country	
21	LocProvinceStateTerritory	stateProvince	
22	LocDistrictCountyShire	county	
23	LocTownship	municipality	
24	LocPreciseLocation	locality	
25	coordinateUncertaintyInMeters	coordinateUncertaintyInMeters	
26	decimalLatitude	decimalLatitude	
27	decimalLongitude	decimalLongitude	
28	verbatimElevation	verbatimElevation	
29	verbatimLatitude	verbatimLatitude	
30	verbatimLongitude	verbatimLongitude	
31	matchRego	matchRego	in_EMu or not_in_EMu

X-CXC = create collection event code

Used with EVENTS-ento reference
Use at the time of creating METHODS

Extract CECs from the summary data
Collection event codes come in square brackets
In column: **SummaryData**
Filter on ^[\[] (regular expression)
Create a new column called **ColEventCode** based on **SummaryData**
value
remove filter

In column: **ColEventCode**
replace(value, "] [", "XXX")
replace(value, /[\]].+$/, "]")
replace(value, "XXX", "] [")

This generates a partial column of collection event codes. (about 9600 in Jan 2013)

X-CEC = Collection Event Code reference

Collection events have Collection Event Codes.
I thought they were supposed to be unique to an event, but they're not. This procedure assumes they are unique to a date.

File is created with Excel and contains Collection Event Codes. Some events appear to have more than one code, so I have extracted the multiple ones out. I'm not sure if that is a good thing?

In Excel, create the file 'EVENTS_CEC reference xx'
Sort so that multiple codes are at the end.
Import into OPENREFINE

Combine dates and Collection Event Codes
Create a new column called **strippedCEC+date** based on **CEC**
cells["ColDateVisitedFrom"].value + " " + fingerprint(value)

Create a new column called **CEC+date** based on **CEC**
cells["ColDateVisitedFrom"].value + " " + value

> This is getting silly… what's the point of a collection event code if you need other stuff?
> Create a new column called **strippedCEC+date+method** based on **CEC**
> cells["ColDateVisitedFrom"].value + " " + fingerprint(value) +
> cells["ColCollectionMethod"].value

Cleaning up the junk and uncertain stuff
replace(value, /^.*(irn).*$/, "")
> replace(value, /^.*\?.*$/, "") doesn't work on a stripped value
> replace(value, /^.*[Ss](ite).*$/, "") – not needed: 'Site 5' is ok if attached to a date

The last few replacements have left blanks. Should we get rid of the blanks entirely?
Probably not necessary.

Get rid of duplicates.
Prefer those with the highest usage
Sort on column: **sequence** – numbers, smallest first (for consistency)
Sort on column: **CountOfUse** – numbers, largest first
Sort on column: **CEC+date** – alphabetical a-z

V-CEC = Collection Event Code cross-check

For BVP stuff. Similar to X-CEC above

CEC analysis is probably optional. Some events are missing CECs so they aren't picked up with the first round of checking against events. This procedure follows the earlier non-CEC analysis, and if any CECs are found to have an IRN they are added to the list of IRNs.

In column: **ColEventCode** Get rid of square brackets outside
First clear the old ones
strip leading and trailing spaces
replace(value, /^[\[\(]/, "")
replace(value, /[\]\)][\s]*$/, "")
replace(value, value[0], toTitlecase(value[0]))
replace(value, value, "["+value+"]") (square brackets aren't needed)

Create a new column called **strippedCEC+date** based on **ColEventCode**
cells["eventDateBegin"].value + " " + fingerprint(value)

Create a new column called **CEC+date** based on **ColEventCode**
cells["eventDateBegin"].value + " " + value

V-CECL = lookup collection event code IRN

For BVP. Finds an IRN for a collection event code

Create a new column called **CEC_irn** based on **strippedCEC+date**
cell.cross("EVENTS_CEC reference 2711a", "strippedCEC+date").cells["event_irn"].value[0]

Muckup

Create a new column called **hasCollection** based on **sequence**
cell.cross("BVP cicadas repaired dates methods 0402a",
"sequence").cells["sequence"].value[0]

Create a new column called **NOTforEMu** based on **catalogNumber**
cell.cross("BVP cicadas NOT for EMu part 8 22Mar2013 1807a",
"catalogNumber").cells["cat_irn"].value[0]

cell.cross("BVP cicadas NOT for EMu 2707a", "catalogNumber").cells["cat_irn"].value[0]

Create a new column called **notValidated** based on **catalogNumber**
cell.cross("BVP validated cicadas baseline 2707a",
"catalogNumber").cells["catalogNumber"].value[0]

V-2SS (OK though probably not needed) = strip site

This procedure tries harder to fit an EVENT irn by fiddling with the SITE (location)
Make sure that locations have got commas where they make sense: viz Manly, Sydney.

Keep a copy of the events. Add to it later.
Create a new column called "**event1_irn**" based on "**event_irn**"
value

Create a new column called "**eventMatch**" based on "**event_irn**"
if(isBlank(value), NULL, "1")
(need to add to this later – after the next set of event_irns)

Rename column: **eventReconstructed** to **eventOKReconstructed** need to return later
Rename column: **strippedLocation** to **strippedOKLocation** need to return later
Create a new column called "**LocOKPreciseLocation**" based on "**LocPreciseLocation**"
value
Create a new column called "**facetLocation**" based on "**LocPreciseLocation**" (may exist)
""

Filter on those locations that haven't got an event_irn
Column: **event_irn** / Facet / Customized facets / Facet by blank / TRUE

Via
Before any more filters:
Column: **LocPreciseLocation**
replace(value, /[\,]?(via)[A-Z][\w]+[\s]?[\w]*$/, "") (remove via …)

Filters
Set up a filter to exclude roads, tracks, rivers from a second pass as these often come with important extra information. eg 'Murray River, near Echuca' is not the same as 'Murray River', and 'Echuca, Murray River' is not the same as 'Echuca'.

Column: **LocPreciseLocation** filter=
\b(River)\b|\b(creek)\b|\b(brook)\b|\b(stream)\b|\b(lagoon)\b|\b(lake[s]?)\b|\b(swamp)\b|\b(dam)\b|\b(watercourse)\b|\b(road)\b|\b(Track)\b|\b(trail)\b|\b(Highway)\b|\b(Drive)\b|\b(Park)\b|\b(District)\b|\b(Forest)\b|\b(Island)|\b(Range[s]?)\b|\b(Mountains)\b|\b(Tableland)|\b(area)\b|\b(cave)

Column: **facetLocation**
"extensive feature"

Column: **LocPreciseLocation**

- remove long filter

Column: **facetLocation**

- text facet / EXCLUDE "extensive feature" – we have now isolated rivers, roads, etc

Column: **LocPreciseLocation**

remove one or two words only. don't remove long narratives (eg near the turnoff to …)

remove only Place names – must begin with a capital letter
replace(value, /[\,]?(near)[A-Z][\w]+[\s]?[\w]*$/, "") (remove “near xx”; eg “near Dubbo”)

New filter – this time without mountains or forest
Column: **facetLocation** clear filter
""

Column: **LocPreciseLocation**
\b(River)\b|\b(creek)\b|\b(brook)\b|\b(stream)\b|\b(lagoon)\b|\b(lake)\b|\b(swamp)\b|\b(dam)\b|\b(watercourse)\b|\b(road)\b|\b(Track)\b|\b(trail)\b|\b(highway)\b|\b(Drive)\b|\b(street)\b|\b(avenue)\b|\b(St)\b|\b(Broadway)\b|\b(Park)\b|\b(District)\b|\b(Island)|\b(Range[s]?)\b|\b(area)\b|\b(cave)\b|\b(Botanic)|\b(Domain)\b

Column: **facetLocation** add a description
"extensive feature"

replace(value, /(, Sydney)*$/, "")
replace(value, /[\,]?(Blue Mountains)$/, "")

Remove column: **facetLocation**

Mount Tamborine

Mount Tamborine and Tamborine Mountain are used interchangeably. In effect they are the probably the same place, so let the events match

Homogenise Tamborine Mountain and Mount Tamborine
Note: in column **event_irn**, keep filter – we are still working with those without an IRN
Column: **LocPreciseLocation**
replace(value, /(Tamborine Mountain)/, "Tamborine Mount") (fudge)

Clean up

remove filter Column: **event_irn** unselect TRUE
Rename column: **event_irn** to **event1_irn**

rename
now run JSON P-XC…

… then return column names in JSON V-R

V-2Xold (ok but now replaced) = mark second pass event matches as being #2

Run this after running 'V-2SS strip site' which finds more events through doing a second pass

Move the first pass event IRNs into the latest list of IRNs
In column: **event_irn**
cells["event1_irn"].value (On error KEEP ORIGINAL)
Remove column **event1_irn**

Mark the latest events as being from a second pass

In column: **eventMatch**
Filter on blanks: eventMatch / Facet / Customized facets / Facet by blank / TRUE

In column: **event_irn**
Filter on non-blanks: event_irn / Facet / Customized facets / Facet by blank / FALSE

In column: **eventMatch**
"2" – corresponds to this event being found in a second pass

X-SS2+ (I don't think this is used now) = relabel second pass events

Run this after running 'V-2SS strip site' which creates a second pass of locations

Originally I had

Rename column: **strippedLocation** to **stripped2Location**
Rename column: **strippedOKLocation** to **strippedLocation**
Rename column: **LocPreciseLocation** to **Loc2PreciseLocation**
Rename column: **LocOKPreciseLocation** to **LocPreciseLocation**

…or try this after running X
Rename column: **eventReconstructed** to **event2Reconstructed**
Rename column: **strippedEvent** to **stripped2Event**
Rename column: **preferredEvent** to **preferred2Event**

..then rename the previous columns
Rename column: **strippedLocation** to **stripped2Location**
Rename column: **strippedOKLocation** to **strippedLocation**
Rename column: **LocPreciseLocation** to **Loc2PreciseLocation**
Rename column: **LocOKPreciseLocation** to **LocPreciseLocation**
Remove: **LocTownship** (don't bother to keep)
Rename column: **LocOKTownship** to **LocTownship**