HOW TO GENERATE THE DIAGRAM METADATA REPORTS

As of March 2019, we have been unsuccessful in getting the “Notes” field to properly export from EA to csv (to ultimately be an Excel spreadsheet). The copy/paste process doesn’t properly handle blank lines within the Notes field and puts each piece of the Notes field in its own cell instead of all pieces in one cell. We have searched help information on the EA User Forum, and also posted the question to the Forum, with no success, even though one of the expert user forum people said this issue had been fixed.

So, Wendy has to create an Access db from EA on an old version of EA, which is on an old computer that she has.

How to run an SQL query from within Access:

<put instructions here after release>

How to run an SQL query from within EA:

Edit > Search in Model > change Common Searches in 1st dropdown to My Searches and then click on New Search (+ on magnifying glass)

1) Run the following query first, then run the second query (run it once for ALL diagrams returned in this first query), and then export all the results to Excel, set up filtering across all columns and format it (details in #3 below), and then filter by diagram one at a time, copy the displayed rows for one diagram (including the headings you formatted) and paste them in a separate file and make individual diagram-specific title changes:

SELECT t\_diagram.name as [DIAGRAM NAME]

FROM t\_diagram

WHERE t\_diagram.Stereotype = 'BRIDG Home Page View';

2) Run this query once - returns rows for all diagrams with the stereotype above then follow instructions above for second query:

(WARNING: IF YOU DO ANY EDITS AND DON'T SAVE FIRST, IT EXPORTS QUERY RESULTS WITH PREVIOUSLY SAVED QUERY EVEN IF IT DISPLAYS ROWS FROM EDITE QUERY.

APPARENTLY EXPORT WORKS OFF WHAT YOU SAVED, NOT WHAT IS DISPLAYED.)

SELECT

t\_diagram.name AS [DIAGRAM NAME],

t\_object.name AS [CLASS NAME],

t\_attribute.name AS [ATTRIBUTE NAME],

t\_attribute.type AS [DATA TYPE],

t\_attribute.lowerbound+'..'+t\_attribute.upperbound AS [CARDINALITY],

t\_attribute.Notes AS [DESCRIPTION]

FROM ((t\_diagramobjects

INNER JOIN t\_object ON t\_object.Object\_ID = t\_diagramobjects.Object\_ID)

INNER JOIN t\_diagram ON t\_diagram.Diagram\_ID = t\_diagramobjects.Diagram\_ID)

INNER JOIN t\_attribute ON t\_attribute.Object\_ID = t\_object.Object\_ID

WHERE t\_object.Object\_Type = 'Class'

AND (

(

(INSTR(t\_diagram.StyleEx,MID(t\_object.ea\_guid,2,6)) = 0)

AND (INSTR(t\_diagramobjects.ObjectStyle,'AttPro=0;AttPri=0;AttPub=0;AttPkg=0') = 0)

AND (INSTR(t\_diagramobjects.ObjectStyle,'AttCustom=0') = 0)

AND "COMMENT: this set brings back attributes for classes where nothing is hidden."

)

OR (

(INSTR(t\_diagramobjects.ObjectStyle,'AttPro=0;AttPri=0;AttPub=0;AttPkg=0') > 0)

AND (INSTR(t\_diagram.StyleEx,MID('S\_'+t\_object.ea\_guid,2,6)) > 0)

AND (INSTR(t\_diagram.StyleEx,MID(t\_attribute.ea\_guid,2,6)) = 0)

AND "COMMENT: this set evaluates to false so that it excludes attributes for classes where all attributes are hidden."

)

OR (

(INSTR(t\_diagram.StyleEx,MID(t\_attribute.ea\_guid,2,6)) = 0)

AND (INSTR(t\_diagram.StyleEx,MID(t\_object.ea\_guid,2,6)) > 0)

AND (INSTR(t\_diagramobjects.ObjectStyle,'AttCustom=0') > 0)

AND "COMMENT: this set shows unhidden attributes - if AttCustom=0 and attribute is NOT IN StyleEx then show the attribute."

)

OR (

(INSTR(t\_diagram.StyleEx,MID(t\_attribute.ea\_guid,2,6)) > 0)

AND (INSTR(t\_diagram.StyleEx,MID(t\_object.ea\_guid,2,6)) > 0)

AND (INSTR(t\_diagramobjects.ObjectStyle,'AttCustom=1') > 0)

AND "COMMENT: this set shows unhidden attributes - if AttCustom=1 and attribute IS IN StyleEx then show the attribute."

)

)

UNION ALL

SELECT

t\_diagram.name AS [DIAGRAM NAME],

t\_object.name AS [CLASS NAME],

'' AS [ATTRIBUTE NAME],

'' AS [DATA TYPE],

'' AS [CARDINALITY],

t\_object.Note AS [DESCRIPTION]

FROM ((t\_diagramobjects

INNER JOIN t\_object ON t\_object.Object\_ID = t\_diagramobjects.Object\_ID)

INNER JOIN t\_diagram ON t\_diagram.Diagram\_ID = t\_diagramobjects.Diagram\_ID)

WHERE t\_object.Object\_Type = 'Class'

ORDER BY 1, 2, 3;

3) Here is the formatting to apply to the initial all-diagrams report and individual reports as appropriate (do as much as possible in the all-diagrams version because then when you copy it into a diagram-specific file it will already have most of the formatting done) - see the existing reports as examples to follow re column widths, etc.:

- Export the report to Excel using the following naming convention: BRIDG<version#>-<YYYYMMDD>-<DiagramNameNoSpacesNoSpecialChars>.xlsx

e.g. BRIDG5.3Beta-20190204-BRIDGBackBone.xlsx (replace "&" with "and", keep "." as is)

- Format it by adding a title row as "BRIDG <Diagram Name> Metadata Report, generated on <Mon. Day, Year>" in Bold font, size 16

- Add another row for the BRIDG URL: "BRIDGmodel.nci.nih.gov" in Bold font, size 14

- (leave out any further header info for now since it may change each time the model is published; keeping it simple)

- Bold the column heading row

- Widen columns so all data shows except in the DESCRIPTION column which needs to be widened AND have word wrap turned on

- Align data at top of all cells

- Freeze the top 3 rows (title, URL, headings)

- Set up filtering across all columns for all data rows (omitting title lines of course)

- Then upload the file to GitHub in a version-specific folder: bridg-model / Documents / Diagram Metadata Reports / BRIDG5.3Beta-20190205 (for example) or BRIDG5.3 (for final version)

- CONSIDER ADDITIONAL FORMATTING: 1) setting a fill color for the column heading cells; 2) turning on display of the cell borders