'**VBA Code** for **Excel2GED3.20.1.15** [**<** update w/current ***filename.xlsm***]**.** '**Print Margins:** T=**0.4**; B=**0.3**; L=**0.5**; R=**0.5"**

'**Edit here** [**NO** hard page breaks]**; Open (Alt+F11)** VBAProject(*filename*); **Copy+Paste** [discards formatting] **into "Genealogy"** Module.

'**Update HEADer (below) with the correct VERSion number of the SOURce (Excel2GED).**

**Option Explicit** 'Option Explicit: Forces declaration of all variables in order to catch/warn of errors; recommended.

Dim **sFile** As String '**sFile**: array variable for GEDCOM output file. **String**: Text.

Dim **nIRowsCnt** As Long, **nUniqInds** As Long, **nUniqFams** As Long, **nFLinkCnt** As Long, **i** As Long 'Large integers; can be > +/-32767.

'**nIRowsCnt**: Rows used; **nUniqInds**: Unique Indiv's; **nUniqFams**: Unique Fam's; **nFLinkCnt**: Relation links.

'--------------------------- '**Update** column/row numbers in **Init()** if cells for **nIRowsCnt, nUniqInds, nUniqFams,** **sFile** are moved.

'--------------------------- '**Update** the following **Const**ant values for columns/column headers if their sequences are changed.

'Const nc**FileNameLoc** = 7 'Column location [7 = **G**] in **row 2** for …:\*path*\*filename*. [**No longer used; see "sFile = " in Init().**]

Const **wrks1row0** = 4 'Individuals' **data begins** on this **row** on wrks**1** ["**Individuals**"].

Const **wrks2row0** = 2 'Family Group **data begins** on this **row** on wrks**2** ["**FamilyGroups**"].

Const **NoteLenMax** = 72 '**Max**imum character length of exported NOTE & SOUR text **segments** (NoteLenMax **+ 1** if last segment),

'including 7-character "**1 NOTE** ", "**1 SOUR** ", or "**2 CONC** " if leading NOTE/SOUR tag is **VBA-supplied**.

'If leading tag is **user-supplied**, "**1** " and any "**2 CONC** " are not included in segment character count.

'**-----------------------------------------------------------------------------------------** 'Column sequences on worksheet**1 (wrks1)**

Const nc**Count1** = **19** '**Count** of wrks**1** **columns** **with** user-entered **Individuals'** **data** used to create GEDCOM. '**@PHOTO-CHANGE-1**

Const nc**Individual** = 1 'A

Const nc**GivenName** = 2 'B

Const nc**LastName** = 3 'C

Const nc**NickName** = 4 'D

Const nc**Sex** = 5 'E

Const nc**FatherID** = 6 'F

Const nc**MotherID** = 7 'G

Const nc**SpouseID** = 8 'H

Const nc**BirthDate** = 9 'I

Const nc**BirthPlace** = 10 'J

Const nc**MarriageDate** = 11 'K

Const nc**MarriagePlace** = 12 'L

Const nc**DeathDate** = 13 'M

Const nc**DeathPlace** = 14 'N

Const nc**PhotoFilePath** = 15 'O '**Column added for ncPhotoFilepath**. '**@PHOTO-CHANGE-2**

Const nc**SourcesNotes** = 16 'P 'Changed from 15 to 16, and O to P, when column added for ncPhotoFilepath; etc. for others.

'Const nc**FatherListed** = -- '- 'v2.4Beta: nc**Fa**ther**ID** column used instead; **ncFatherListed** column **redundant**; v3.0: **deleted**.

'Const nc**MotherListed** = -- '- 'v2.4Beta: nc**Mo**ther**ID** column used instead; **ncMotherListed** column **redundant**; v3.0: **deleted**.

'Const ncSpouseListed = -- '- 'v2.4Beta: nc**Sp**ouse**ID** column used instead; **ncSpouseListed** column **replaced** with **ncChildren**.

Const nc**ParentFamilyID** = **17** '**Q** '**Own**FamID **of parent**(s), **assigned on wrks2**, then copied to **wrks1** Parent's Family ID column.

Const nc**Children** = **18** '**R** 'v2.4Beta: nc**Children** column (header: '**Child**-**ren?**') replaced the nc**SpouseListed** column.

Const nc**OwnFamilyID** = **19** '**S** '**Derived by** "OwnFamilyID" **formula** for Couples **and/or** Single Parents.

'**-----------------------------------------------------------------------------------------** 'Column sequences on worksheet2 **(wrks2)**

Const nc**Count2** = **5** '**Count** of **columns** of **FamilyID2 table** ofsheet**2** (**wrks2**).

'-------------------------------------------------------------------------------------------- '**List** of **Linked** Individuals of wrks**2**

Const nc**FamilyID2** = 1 'For "**OwnFamilyID**" derived by formula on wrks**1** for Couples and/or Parent(s). Initially **un**sorted.

Const nc**Individual2** = 2 'For **Individual ID**s read from wrks**1**.

Const nc**Relation2** = 3 'For family links. Using "**aH**", "**bW**", "**cC**" vs. "H", "W", "C", respectively, for traditional sorting.

Const nc**MarriageDate2** = 4 'For Marriage Dates read from wrks**1**.

Const nc**MarriagePlace2** = 5 'For Marriage Places read from wrks**1**.

'-------------------------------------------------------------------------------------------------------------- '**Table** of **Families**

Const nc**FamilyID3** = 7 'For **wrks2** table of **Family Groups**: **Couples** and/or **(single) Parent**(s) with **OwnFamilyID**, from wrks**1**.

Const nc**HusbID** = 8 'For wrks**2** table Husband IDs.

Const nc**WifeID** = 9 'For wrks**2** table Wife IDs.

'-------------------------------------------------------------------------------------------------------------- '**Column**: **ParFamID**s

Const nc**ParFamID** = **17** 'For wrks**2** column for Parent's Family ID = Parent's OwnFamilyID. 'Col **Q**.

'---------------------------------------------------------------------------- '**Submitter** variables for worksheet3 **(wrks3) Template**

Dim s**SubmName** As String 'Submitter's **Name**

Dim s**SubmAddr** As String 'Submitter's **Street Address**

Dim s**SubmCity** As String: Dim s**SubmState** As String: Dim s**SubmPostal** As String 'Submitter's **City, State, Postal Code**

Dim s**SubmCtry** As String 'Submitter's **Country**

Dim s**SubmPhone** As String 'Submitter's **Telephone Number**

Dim s**SubmEmail** As String 'Submitter's **Email Address**

Dim **wrks1** As Worksheet, **wrks2** As Worksheet, **wrks3** As Worksheet '------------------------------------------ 'Worksheets with data

**Private Sub Init()** '**===========================================** 'Update references if rename wrks or add/delete/move row/columns.

Set **wrks1** = ThisWorkbook.Worksheets("**Individuals**") 'Data for individuals and their links to others entered on wrks**1**.

Set **wrks2** = ThisWorkbook.Worksheets("**FamilyGroups**") 'Linked individuals organized by VBA code into families on wrks**2**.

Set **wrks3** = ThisWorkbook.Worksheets("**Submitter**") 'v3.0: Added user entry form for Submitter information.

s**File** = wrks**1**.Cells(wrks1row0 **- 3**, **7**) '**Path\Filename** location for GEDCOM output. See **btnBrowse\_Click()** and **ValidFile()**.

**nIRowsCnt** = wrks**1**.Cells(wrks1row0 **- 3**, **1**) '**I**ndiv-**Rows** **C**ou**nt** in a cell **3** lines above first data row.

**nUniqInds** = wrks**1**.Cells(wrks1row0 **- 2**, **1**) 'was Cells(**2**, **1**) 'Unique Individuals = MAXimum / **highest Individual ID number**.

**nUniqFams** = wrks**1**.Cells(wrks1row0 **- 2**, **3**) 'was Cells(**2**, **3**) 'Unique Families = MAXimum / **highest OwnFamID number**.

End Sub '**End Init() =============================================================================================================**

**Private Function ExportGed() As Boolean '=======================** 'Boolean: **ExportGed = True if** Function completes w/o error. **=====**

Dim s**Ged** As String 'P.S.: Removed in v3.0: "Dim … , **cnt** As Long" [was never used].

Dim n**InformFreq** As Long, b**Feedback** As Boolean, n**Counter** As Long

Dim i**Front** As Long, i**Back** As Long, i**Len** As Long, i**Note** As Long, i**Source** As Long 'Array variables for @SourcesNotes.

Dim i**PrevChild** As Long 'Most recent child processed in organizing a family group.

s**SubmName** = wrks3.Cells(2, 2) 'Submitter's Name

s**SubmAddr** = wrks3.Cells(3, 2) 'Submitter's Address

s**SubmCity** = wrks3.Cells(4, 2) 'Submitter's City

s**SubmState** = wrks3.Cells(5, 2) 'Submitter's State 'Added per GEDCOM 5.5.1 spec.

s**SubmPostal** = wrks3.Cells(6, 2) 'Submitter's Postal Code 'Added per GEDCOM 5.5.1 spec.

s**SubmCtry** = wrks3.Cells(7, 2) 'Submitter's Country

s**SubmPhone** = wrks3.Cells(8, 2) 'Submitter's Telephone

s**SubmEmail** = wrks3.Cells(9, 2) 'Submitter's Email

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* '**Change Feedback Frequency Here** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

b**Feedback** = True 'Feedback: Used in "Individual Record" and "Family Record" sections.

n**InformFreq** = 1000 'For every **nInformFreq** records added to GEDCOM, report current count **in** VBA screen **Immediate Window**.

**On Error GoTo Error\_Handler** '**On Error:** "Error during Export" displayed; sFile closed.

Call **Init** '**Initialize:** wrks1, wrks2, wrks3, sFile, nIRowsCnt, etc.

Open **sFile** For Output As **#1**  'Allow string data to be written to the "sFile" file.

**'Header** '---------------------------------------------------------- '**Begin**ning lines of the GEDCOM file.

sGed = "" 'Initialize/clear sGed string variable.

sGed = sGed + "0 HEAD" & vbCrLf 'Add first line to sGed variable.

sGed = sGed + "1 SOUR **Excel2GED.xlsm**" & vbCrLf 'Was: "… PAF"

sGed = sGed + "2 **VERS** **3.20.1.15**" & vbCrLf '**< Update as needed <<** 'Was: "… 5.2.18.0" '**No leading zeros** in Version pieces.

sGed = sGed + "2 NAME **Excel2GED**" & vbCrLf 'Was: "… Personal Ancestral File"

sGed = sGed + "2 CORP **SourceForge.net**" & vbCrLf 'Was: "… The Church of Jesus Christ of Latter-day Saints"

sGed = sGed + "3 ADDR" & vbCrLf '**GEDCOM 5.5.1**: No line value. 'Was: "3 ADDR 50 East North Temple Street"

sGed = sGed + "4 ADR1 **World Wide Web"** & vbCrLf 'Was: "4 CONT Salt Lake City, UT 84150"

sGed = sGed + "4 ADR2 **Internet**" & vbCrLf 'Was: "4 CONT USA" 'GEDCOM 5.5.1: **ADRn** replaces **CONT**

sGed = sGed + "3 WWW **sourceforge.net/projects/excel2ged**" & vbCrLf '**GEDCOM 5.5.1**: Use **n WWW** tag for **URL**, same level as **ADDR**

sGed = sGed + "1 DEST **Other**" & vbCrLf 'Intended/Expected Destination for this GEDCOM file.

sGed = sGed + "1 **DATE** " & Format(Date, "dd mmm yyyy") & vbCrLf 'File Creation Date.

sGed = sGed + "2 **TIME** " & Format(Time, "hh:mm:ss") & vbCrLf 'File Creation Time.

sGed = sGed + "1 SUBM @U1@" & vbCrLf 'Link to Submitter record (see below).

sGed = sGed + "1 FILE " & sFile & vbCrLf 'Name of GEDCOM file.

sGed = sGed + "**1** GEDC" & vbCrLf 'GEDCOM this is; details follow.

sGed = sGed + "**2** VERS **5.5.1**" & vbCrLf 'GEDCOM version; was: "… **5.5**"

sGed = sGed + "**2** FORM LINEAGE-LINKED" & vbCrLf 'LINEAGE-LINKED: " … to specify GEDCOM compatible … "

sGed = sGed + "1 **CHAR UTF-8**" & vbCrLf 'v3.0: Character Set changed back to (specified) **UTF-8** vs. ASCII.

sGed = sGed + "1 LANG English" & vbCrLf 'Language used.

**Print #1, sGed;** '----------------------------------------- 'Append Header info in "sGed" array variable to file #1 ("sFile").

sGed = "" 'Initialize sGed string array variable. 'Begin **Submitter Record**.

sGed = sGed + "0 @U1@ **SUBM**" & vbCrLf 'v3.0: **SUBMitter** vs. obsolete "Submission" (SUB**N**) record (for AF).

sGed = sGed + "1 NAME " & sSubmName & vbCrLf 'Submitter's name, **from wrks3**.

sGed = sGed + "**1** **ADDR** " & vbCrLf 'Was: "1 ADDR " **& sSubmAddr** 'But **GEDCOM 5.5.1**: "No line value".

sGed = sGed + "**2** **ADR1** " & sSubmAddr & vbCrLf 'Submitter's **Street** address, per wrks3.

sGed = sGed + "**2** CITY " & sSubmCity & vbCrLf 'Submitter's **City**, per wrks3.

sGed = sGed + "**2** **STAE** " & sSubmState & vbCrLf 'Submitter's **State**, per wrks3.

sGed = sGed + "**2** **POST** " & sSubmPostal & vbCrLf 'Submitter's **Postal Code**, per wrks3.

sGed = sGed + "**2** CTRY " & sSubmCtry & vbCrLf 'Submitter's **Country**, per wrks3. 'Added: "2 STAE" & "2 POST".

sGed = sGed + "**1 PHON** " & sSubmPhone & vbCrLf 'Submitter's **Telephone** number, per wrks3.

sGed = sGed + "**1 EMAIL** " & sSubmEmail & vbCrLf 'Submitter's **Email** address, per wrks3.

**Print #1, sGed;** '----------------------------------------- 'Append what is in "sGed" (Submitter record) to #1 ("sFile") file.

**'--------------- CreateFam() organizes Linked Individuals into Family Groups on wrks2 --------------**

Call **CreateFam** '**Gather** Individuals having family links to others; **sort** by OwnFamID; **group** as families in new table.

'**---------------------------------------------------------------------------------------------------**

**'Individual Record '-------------------------------------- 'Add "Individual" records to GEDCOM. -----------------------------**

ReDim sColumn(**1** To **ncCount1**) As String 'ReDim: reinitialize the array.

Dim **col** As Integer '"col" used to temporarily designate **col**umns of Indiv's data.

Dim **LastINDI** As String, **sOBJ** As String '"LastINDI" string variable: most recently processed Individual.

**LastINDI** = "": **sOBJ** = "" 'Initialize LastINDI, sOBJ. 'sOBJ collects MM records for later.

**nCounter** = 0 'Initialize the loop counter.

**For i** = wrks1row0 To **nIRowsCnt +** **wrks1row0 - 1** '---------- 'Last row # = total entries + non-data rows preceding data rows.

DoEvents '"DoEvents allows the Operating System to process events while a VBA program is running."

nCounter = nCounter + 1 'Increment the counter.

If (b**Feedback** And (n**Counter** Mod n**InformFreq** = 0)) Then Debug.Print "Individual Count: " & nCounter '**Mod**: div'n remainder.

'**Display** current **loop count** whenever **= nInformFreq.**

**On Error Resume Next** '**#N/A** may cause an error, so **skip row if data is missing**.

'-------------------------------------------------------------------------------------------------------------------------

For **col** = 1 To **ncCount1** '**Read** user-entered **data of Indiv** in columns of **row i** into **sColumn**.

sColumn(**col**) = Trim(wrks1.Cells(**i**, **col**)) 'Trim: Remove leading and trailing spaces.

Next 'Read data of next cell to the right into the sColumn array.

On Error GoTo Error\_Handler '**On Error:** "Error during Export" displayed; sFile closed.

sGed = "" 'Initialize sGed.

**If** sColumn(ncIndividual) **<> LastINDI** Then '**If** this row is now for a **different Individual**, **Then** …

sGed = sGed + "**0 @I**" & sColumn(ncIndividual) & "**@ INDI**" & vbCrLf

sGed = sGed + "**1 NAME** " & sColumn(nc**GivenName**) & "**/**" & sColumn(nc**LastName**) & "**/**" & vbCrLf

sGed = sGed + "**2 GIVN** " & sColumn(nc**GivenName**) & vbCrLf

sGed = sGed + "**2 SURN** " & sColumn(nc**LastName**) & vbCrLf

If sColumn(ncNickName) <> "" Then sGed = sGed + "**2 NICK** " & sColumn(nc**NickName**) & vbCrLf 'v3.0: **NICK** added to Export.

If sColumn(ncSex) **<> ""** Then sGed = sGed + "**1 SEX** " & sColumn(nc**Sex**) & vbCrLf 'If stated, add to GEDCOM.

If sColumn(ncBirthDate) <> "" Or sColumn(nc**BirthPlace**) <> "" Then

sGed = sGed + "**1 BIRT**" & vbCrLf

If sColumn(ncBirthDate) <> "" Then sGed = sGed + "**2 DATE** " & sColumn(nc**BirthDate**) & vbCrLf

If sColumn(ncBirthPlace) <> "" Then sGed = sGed + "**2 PLAC** " & sColumn(nc**BirthPlace**) & vbCrLf

End If

If sColumn(ncDeathDate) <> "" Or sColumn(ncDeathPlace) <> "" Then

sGed = sGed + "**1 DEAT**" & vbCrLf

If sColumn(ncDeathDate) <> "" Then sGed = sGed + "**2 DATE** " & sColumn(nc**DeathDate**) & vbCrLf

If sColumn(ncDeathPlace) <> "" Then sGed = sGed + "**2 PLAC** " & sColumn(nc**DeathPlace**) & vbCrLf

End If

**End If** 'The Indiv info above only included for 1st row of them; skip for add'l occurrences.

**LastINDI** = sColumn(ncIndividual) '**LastINDI:** Individual of row just processed; they possibly appear again in next row.

**'Spouse family - FAMS record** 'To an Individual's GEDCOM record, **add their OwnFamID** number, if available.

If sColumn(nc**OwnFamilyID**) <> "" Then sGed = sGed + "**1 FAMS @F**" & sColumn(nc**OwnFamilyID**) & "**@**" & vbCrLf '**Family Group no.**

'GEDCOM spec example lists FAM**C** before FAM**S**, but AQ, LFT, RM all export FAM**S** first.

**'Parent's family - FAMC record '**To an Individual's GEDCOM record, **add their parent's ParFamID number**, if available.

If sColumn(nc**ParentFamilyID**) <> "0" Then

sGed = sGed + "**1 FAMC @F**" & sColumn(nc**ParentFamilyID**) & "**@**" & vbCrLf '**FAMC numbers are what link generations.**

End If

**'MsgBox** "Parent's family ID (FAMC) + Spouse/OwnFamID family (FAMS) have been added for this row." '**Enable for Debugging << ----<<**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* '**ADD EXTENDED FIELD'S CODE HERE** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'**@PHOTO-CHANGE-3a** --- **"MULTIMEDIA\_LINK"** --- Imbedded with Individual to whom the item is attached ------------------------

If sColumn(nc**PhotoFilePath**) <> "" Then '**If** PhotoFilePath **present**: Then to an Individual's GEDCOM record **add …**

'**MsgBox** "Photo File Path: " & sColumn(ncPhotoFilePath) 'Enable for Debug of @PhotoFilePath section. << --------<<

sGed = sGed + "**1 OBJE** @**M**" & **sColumn(ncIndividual) & "000" & i & "@"** & vbCrLf '**MULTIMEDIA\_LINK** to Individual's photo.

'-------------------------------------------------------------------------------------------------------------------------

'**@PHOTO-CHANGE-3b --- "MULTIMEDIA\_RECORD" ---** MM/OBJE records must be added outside the Individual(s) section.

sOBJ = sOBJ + "**0 @M" & sColumn(ncIndividual) & "000" & i & "@ OBJE**" & vbCrLf '"**<XREF:OBJE>":** "**M**" + Ind's **ID&Row**#.

sOBJ = sOBJ + "**1 FILE** " & sColumn(nc**PhotoFilePath**) & vbCrLf 'Was: "**2 FORM** jpeg" … '**FILE** / **FORM** lines swapped.

sOBJ = sOBJ + "**2 FORM jpg**" & vbCrLf '**Change** **if not** **jpg**. 'Was: "**1 FILE** " **…** '**GEDCOM 5.5.1**: FORM **after** FILE.

sOBJ = sOBJ + "**2 TITL** M" & **sColumn(ncIndividual) & "000" & i** & vbCrLf '**sPhotoTitle** (of a new column).

'These 0-level MM/OBJE records must be added to sGed after/outside the 0-level Individual section, which includes Sources & Notes.

End If

'---------------------------------------------------------------------------------------------------------------------------------

'**@SourcesNotes** --------------------------------- '**Text from Sources ("SOUR ") & Notes ("NOTE ") column** -------------------

If sColumn(ncSourcesNotes) <> "" Then '**If** Sources or Notes **present**: Then to an Individual's GEDCOM record **add …**

i**Front** = 1 'i**Front** = **First** character position of this row's Sources & Notes entry.

i**Len** = **Len**(sColumn(ncSourcesNotes)) 'i**Len** = **Len**gth of text string in this row's Sources & Notes entry/cell.

i**Back** = i**Len** 'i**Back** = **Last** character position of this row's Sources & Notes entry.

i**Note** = **InStr**(iFront, sColumn(ncSourcesNotes), "**NOTE** ", vbBinaryCompare) 'Find start of 1st "**NOTE** " keyword.

i**Source** = **InStr**(iFront, sColumn(ncSourcesNotes), "**SOUR** ", vbBinaryCompare) 'Find start of 1st "**SOUR** " keyword.

'vbBinaryCompare: Exact match.

If iNote = 0 And iSource = 0 Then 'If **no key words: all one note, Then…**

sGed = sGed + "**1 NOTE** " 'Add keyword start for GEDCOM note.

sGed = sGed + **WriteNotes**(sColumn(nc**SourcesNotes**), iFront, iBack) 'Append note text to Indiv's record.

iFront = iBack 'This was a single, unkeyed note: set Front/First character = Last = **done**.

ElseIf **iNote <> 1 And iSource <> 1** Then '**Leading unkeyed** NOTE **followed by** at least one **keyed** NOTE **or** SOUR.

If (iNote = 0) Then 'If no "NOTE " keyword found, Then process first "SOUR " text.

**iFront** = **iSource** 'Set iFront = character position of "**S**" of first "**S**OUR " keyword.

ElseIf (iSource = 0) Then 'If no "SOUR " keyword found, Then process first "NOTE " text.

**iFront** = **iNote**  'Set iFront = character position of "**N**" of first "**N**OTE " keyword.

Else 'Both SOUR and NOTE keywords follow leading unkeyed NOTE, so:

**iFront** = WorksheetFunction.**Min**(iNote, iSource) 'Set "iFront" = position of **first character** of **next** keyword.

End If 'WorksheetFunction.**Min(. . .)**: return **Min**imum value from list.

'

sGed = sGed + "**1 NOTE** " '**Keyed** notes & sourcesonly need "**1** " inserted at their beginning.

sGed = sGed + **WriteNotes**(sColumn(nc**SourcesNotes**), 1, **iFront - 1**) '**Send** text (first character **up to** **one position**

' **before** keyword) to WriteNotes() for parsing.

'

End If 'Now there's at least one NOTE or SOUR, beginning at iFront, or else iFront = iBack = done.

'Continued (@SourcesNotes) . . .

'Blank lines added to adjust soft (vs. hard) page print breaks. (Hard break code is not fully stripped when pasted into VBA code.)

**While iFront < iLen** 'Any **unkeyed** NOTE **preceding** keyed NOTE(s)/SOUR(s) has been appended to sGED;

'**now process those that are keyed until all of ncSourcesNotes is consumed.**

**iNote** = InStr(**iFront + 4**, sColumn(ncSourcesNotes), "**NOTE** ", vbBinaryCompare) 'Look starts after current keyword.

'Find **start pos** of **next** "**NOTE** " (if any); start search for next keyword right after previous "NOTE " keyword.

**iSource** = InStr(**iFront + 4**, sColumn(ncSourcesNotes), "**SOUR** ", vbBinaryCompare) 'Look starts after current keyword.

'Find **start pos** of **next** "**SOUR** " (if any); start search for next keyword right after previous "SOUR " keyword.

sGed = sGed + "**1** " 'A "**1** " **must precede** required keywords, "**NOTE** " & "**SOUR** " (note the spaces).

'"**NOTE** " / "**SOUR** " come at the beginning of each of their text segments in ncSourcesNotes.

If iNote = 0 And iSource = 0 Then 'No "NOTE " nor "SOUR " found: Must have **consumed the last** NOTE or SOURce.

**iBack** = **iLen** 'Set i**Back** = **Len**gth of 'ncSourcesNotes' **so the** "While iFront < iLen" **loop**

'**will end this time** due to "iFront = iBack".

sGed = sGed + **WriteNotes**(sColumn(ncSourcesNotes), iFront, iBack)

ElseIf iNote = 0 Or iSource = 0 Then '**Failing preceding test,**

'**test IF one keyword BUT NOT the other still found; if TRUE, then …**

**iBack** = WorksheetFunction.**Max**(iNote, iSource) 'Text string ends **w/FIRST** CHARACTER POSITION of **N**OTE or **S**OUR,

'whichever WAS found present in preceding test(s).

sGed = sGed + WriteNotes(sColumn(ncSourcesNotes), iFront, **iBack - 1**)

'"**iBack-1**" removes first character of **N**OTE or **S**OUR included by preceding line of VBA code.

Else 'Otherwise, …

**iBack** = WorksheetFunction.**Min**(iNote, iSource) '**Both** NOTE & SOUR present:

'Text string ends **w/FIRST** CHARACTER POSITION of whichever is next,

'**N**OTE or **S**OUR, but that character is removed by next line of code:

sGed = sGed + **WriteNotes**(sColumn(ncSourcesNotes), iFront, **iBack - 1**)

'"**iBack-1**" removes first character of **N**OTE or **S**OUR included by preceding line of VBA code.

End If '

iFront = iBack 'Proceed to next segment.

'**Reducing iBack in the wrong place will end the loop prematurely due to "While iFront < iLen".**

**W**end 'All of this row's 'ncSourcesNotes' consumed/added to sGED.

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'v3.0: **CHAN/DATE/TIME of v2.4 deleted; not applicable in Excel2GED. See File Creation DATE and TIME in the header.**

'**GEDCOM 5.5.1:** "**The change date is intended to only record the last change to a** [specific Individual's] **record.**"

' Excel2GED does not have the ability to uniquely date stamp changes to individual records.

' Also, v2.4: "\_CHAN", "\_DATE", "\_TIME" are non-standard GEDCOM tags, resulting in errors upon import.

'-------------------------------------------------------------------------------------------------------------------------

**Print #1, sGed;** '----------------------------------------- '**End this GEDCOM entry for this row for an Individual.**

**Next i** '----------------------------------------- 'Process next row (**may be same Indiv, different family group**).

**Print #1, sOBJ;** 'Append **OBJE** records gathered in the MULTIMEDIA\_**RECORD** section that follows the MULTIMEDIA\_**LINK** section.

**'Family Record '------------------------------------------ 'Add "Family Group" records to GEDCOM.**

Dim **pFamID** As Long, **mRecord** As Boolean '**pF**am**ID**: **p**revious **Fam**ily **ID**; **mRecord**: Marriage Record.

ReDim sColumn**2**(1 To ncCount**2**) As String 'ReDim: reinitialize the array.

pFamID = -1 'Initializing with "" or "0" could result in error at 1st compare.

nCounter = 0 'Initialize the loop counter.

For **i** = 2 To **nFLinkCnt** + 1 'Loop starts w/ i = [row] 2; row 1 is for column headers.

DoEvents

nCounter = nCounter + 1

If (bFeedback And (nCounter Mod n**InformFreq** = 0)) Then Debug.Print "Family Count: " & nCounter

For **col** = 1 To **ncCount2** 'Read **row i** values of **FamilyID2 table.** Skip if #N/A (empty cell).

**sColumn2**(**col**) **=** Trim(**wrks2**.Cells(**i**, **col**)) 'Trim: Remove leading and trailing spaces.

**Next** 'Read data of next cell to the right into the sColumn**2** array.

If sColumn2(**ncFamilyID2**) <> **pFamID** Then '**If** current FamilyID2 <> **p**revious **Fam**ily**ID**2, **Then** start a new Family group.

sGed = ""

sGed = sGed + "**0 @F**" & sColumn2(**ncFamilyID2**) & "**@ FAM**" & vbCrLf 'First line of a GEDCOM **Family** record.

Print #1, sGed;

**pFamID** = **sColumn2**(**ncFamilyID2**) '

**mRecord** = False 'False: Marriage record not yet added to GEDCOM for this couple.

End If

sGed = "" 'For gen'y s/w compatibility, **second encountered** of same-sex couple **exported** to GEDCOM **in** **opposite gender role:**

If sColumn2(ncRelation2) = "**aH**" Then sGed = sGed + "**1 HUSB @I**" & sColumn2(ncIndividual2) & "**@**" & vbCrLf **'See CreateFam().**

If sColumn2(ncRelation2) = "**bW**" Then sGed = sGed + "**1 WIFE @I**" & sColumn2(ncIndividual2) & "**@**" & vbCrLf **'See CreateFam().**

If sColumn2(ncRelation2) = "**cC**" Then

If sColumn2(ncIndividual2) <> i**PrevChild** Then 'Filter duplicate children, e.g., multiple entries of same person.

sGed = sGed + "**1 CHIL @I**" & sColumn2(nc**Individual2**) & "**@**" & vbCrLf '[Filter nd'd for child w/multiple marriages.]

i**PrevChild** = sColumn2(ncIndividual2) 'Most recent child processed in organizing a family group.

End If

End If

If (sColumn2(ncMarriageDate2) <> "" Or sColumn2(ncMarriagePlace2) <> "") And Not **mRecord** Then

sGed = sGed + "**1 MARR**" & vbCrLf

If sColumn2(ncMarriageDate2) <> "" Then sGed = sGed + "**2 DATE** " & sColumn2(ncMarriageDate2) & vbCrLf

If sColumn2(ncMarriagePlace2) <> "" Then sGed = sGed + "**2 PLAC** " & sColumn2(ncMarriagePlace2) & vbCrLf

**mRecord** = True 'True: Marriage record has been created in GEDCOM for this couple.

End If

Print #1, sGed;

Next **i** '---------------------------------------------------'Process/Create next Family Record. ------------------------------

**'Trailer '------------------------------------------------ 'Section to close the GEDCOM file.**

sGed = ""

sGed = sGed + "**0 TRLR**" & vbCrLf 'The End.

Print #1, sGed;

**Close #1 '------------------------------------------------ 'Closing of sFile (the GEDCOM file).**

ExportGed = True 'GEDCOM export was successful.

**Exit** Function

**Error\_Handler: '---------------------------------------------- 'Gracefully end** the export process if an error is encountered.

MsgBox Err.Description, vbCritical, "**Error during export. Last Ind|Fam in process:** " & **LastINDI** & "**|**" & **pFamID**

**Close #1 'Close sFile if export fails.**

End Function **'End ExportGed() ===================================================================================================**

**Private Sub CreateFam() '========================================================================================================**

'CreateFam(): **wrks2:** **Row** for **each** Indiv's **role** (H, W, C) **in family** [col A:**famid**, col B:**IndivID**, col C:**role**]**; sort; …**

**'**then **regroup into "OwnFamID"** tablew/HusbID &/or WifeID **for those w/partner and/or parent-child links**.

Dim **j** As Long, **famid** As Long

'Clear wrks**2** in case next 'Export to GEDCOM' has fewer Individuals in wrks**1,** resulting in leftovers.

'"**ClearContents**:" leaves formulas and formatting (including borders) vs. "Clear".

wrks2.Range("**A2**:**Z**65536").**ClearContents** 'Clear all data but leave headers of **wrks2**.

' wrks2.Range("**?**2:**?**65536").**ClearContents** 'Enable **for debugging** after customizing this line and previous line, as needed.

wrks2.Cells(1, nc**FamilyID2**) = "**Family ID**" '**Column for OwnFamID, copied from wrks1.**

wrks2.Cells(1, ncIndividual2) = "**Individual**"

wrks2.Cells(1, ncRelation2) = "**Relation**" 'Family role: Using "**aH**", "**bW**", "**cC**" vs. "H", "W", "C" for traditional sorting.

wrks2.Cells(1, ncMarriageDate2) = "**Marriage Date**"

wrks2.Cells(1, ncMarriagePlace2) = "**Marriage Place**"

wrks2.Cells(1, nc**FamilyID3**) = "**OwnFamID**" 'v3.0 **Fam**ily**Gr**ou**p** table: "OwnFamID" vs. "Fam ID" for clarity (it's **not** "ParFamID").

wrks2.Cells(1, ncHusbID) = "**Husb ID**" '**FamGrp table**: one row for each Family Group w/ HusbID &/or WifeID when known.

wrks2.Cells(1, ncWifeID) = "**Wife ID**" '**FamGrp table**: To appear, Indiv must claim a partner or have a child claim them.

wrks2.Cells(1, nc**ParFamID**) = "**ParFam ID**" '**v3.0:** Added missing **column header**; data below it will be copied to wrks**1**.

**Application.ScreenUpdating = False** '**Disables Screen Updates to speed up processing**; re-enabled later**.**

**j** = wrks**2**row0 - 1 'Initialize j to **0**. '**Fill wrks2 FamilyID2 table w/wrks1 Indiv's w/OwnFamID; will use for FamID3 table.**

**On Error Resume Next** '**Skip wrks1 rows w/o OwnFamID entry; i.e., only collect for Individuals with one.**

For **i** = wrks**1**row0 To **nIRowsCnt** + wrks1row0 – 1 'Loop starts w/ **i** = [first row of data on wrks1], ends at last row with data.

**famid** = **wrks1**.Cells(**i**, **ncOwnFamilyID**) '**famid:** from **"Own Family ID"** column of **wrks1**.

If Err = 0 Then 'If no error occurred, Then …

**j** = **j** + 1 'Loop variable "**j**" used for addressing **wrks2 rows** w/ Indiv's of wrks1 w/OwnFamID.

wrks2.Cells(**j**, nc**FamilyID2**) = **famid**

wrks2.Cells(**j**, nc**Individual2**) = **wrks1**.Cells(**i**, nc**Individual**)

'**Fill FamID2 table Relation w/ "aH" & "bW", FamID3 table w/ ID numbers of aH & bW**.

**If** wrks**1**.Cells(**i**, ncSex) = "**F**" Then

**wrks2**.Cells(**j**, nc**Relation2**) = "**bW**"

If **wrks2**.Cells(wrks2row0 + famid - 1, **ncWifeID**) **= ""** Then 'If **WifeID** column **blank**, then **…**

**wrks2**.Cells(wrks2row0 + famid - 1, **ncWifeID**) = wrks1.Cells(**i**, ncIndividual) '**Copy WifeID** to WifeID column**.**

**Else: wrks2**.Cells(**j**, nc**Relation2**) = "**aH**" '**same-sex couple: second one encountered goes in opposite gender role.**

**wrks2**.Cells(wrks2row0 + famid - 1, **ncHusbID**) = wrks1.Cells(**i**, ncIndividual) '**Copy HusbID** to HusbID column,

End If **'& change "bW" to "aH" in Family ID.**

**Else 'If not "F"**, then **assume "M", …**

**wrks2**.Cells(**j**, nc**Relation2**) = "**aH**" '**even if** Sex is **"U"** or **blank**/empty.

If **wrks2**.Cells(wrks2row0 + famid - 1, **ncHusbID**) = "" Then 'If **HusbID** column **blank**, then …

**wrks2**.Cells(wrks2row0 + famid - 1, **ncHusbID**) = wrks1.Cells(**i**, ncIndividual) '**Copy HusbID** to HusbID column.

**Else: wrks2**.Cells(**j**, nc**Relation2**) = "**bW**" '**same-sex couple: second one encountered goes in opposite gender role.**

**wrks2**.Cells(wrks2row0 + famid - 1, **ncWifeID**) = wrks1.Cells(**i**, ncIndividual) '**Copy WifeID** to WifeID column,

End If **'& change "aH" to "bW" in Family ID**

End If

wrks2.Cells(**j**, ncMarriageDate2) = wrks1.Cells(**i**, ncMarriageDate)

wrks2.Cells(**j**, ncMarriagePlace2) = wrks1.Cells(**i**, ncMarriagePlace)

End If

Err = 0 'Reset Err in case any occurred.

Next **i** '---------------------------------------------------------------------------------------------------------------------

For **famid** = 1 To **wrks1**.Cells(2, **ncOwnFamilyID**) '**For** all **ncOwnFamilyIDs** indicated by count on wrks1, **find parents,**

**wrks2**.Cells(wrks2row0 + famid - 1, **ncFamilyID3**) = **famid** 'add their **OwnFamID to ParFamID col., add cC's to FamilyID3 table.**

**Next famid** '**OwnFamID column formulas of wrks1 assign a # for any Individual w/link(s) to a spouse &/or child.**

For **i** = **wrks1**row0 To **nIRowsCnt** + wrks1row0 - 1 '**For all Data Entry Rows** of wrks**1**, **add** their **ParFamID & "cC" IF …**

**If** (wrks1.Cells(**i**, nc**FatherID**) <> "") **Or** (wrks1.Cells(**i**, nc**MotherID**) <> "") Then '**If a parent identified in wrks1:**

For **famid** = 1 To **wrks1**.Cells(**2**, **ncOwnFamilyID**) '**If FamID3 parents match wrks1 parents; use their OwnFamID as own:**

If **wrks2**.Cells(wrks2row0 + famid - 1, nc**HusbID**) **=** **wrks1**.Cells(**i**, nc**FatherID**) Then 'A match includes blank = blank.

If **wrks2**.Cells(wrks2row0 + famid - 1, nc**WifeID**) **=** **wrks1**.Cells(**i**, nc**MotherID**) Then 'A match includes blank = blank.

wrks2.Cells(**i**, **ncParFamID**) = **famid** '**Copy Indiv's ParFamID** (if has parent) to **ParFamID column.**

**j** = **j** + 1 '**Children added** to end of FamilyID table **before sorting by famid.**

wrks2.Cells(**j**, ncFamilyID2) = **famid** '**Found parent's OwnFamID assigned to child entry** in current row.

wrks2.Cells(**j**, ncIndividual2) = **wrks1**.Cells(**i**, nc**Individual**) '**As a child, Individual's ID** entered in current row.

wrks2.Cells(**j**, ncRelation2) = "**cC**" '**Child's role** entered in **Relation** column of current row.

End If

End If

Next **famid**

End If '**Children added to end of FamilyID list; list then sorted per keys.**

Next **i '**------------------------------------------------- 'Next wrks**1** row (not necessarily a new Individual). ---------------

**'Stop '------------------------------------------------------** **'Enable "Stop" to debug CreateFam(). <--------------------------<<**

**wrks2**.Range("**A**" & wrks**2**row0 & "**:E**" & **j** + 1).**Sort** **Key1**:=wrks2.Range("**A**" & wrks2row0), **Key2**:=wrks2.Range("**C**" & wrks2row0) '**SORT**

**Application.ScreenUpdating = True** 'Re-enable Screen Updates. 'v3.0: added **2nd Sort Key**: sort on 'Relation' **after** 'Family ID'.

**nFLinkCnt** = **j** - wrks**2**row0 + 1 '**Relation Links**: Parent-to-Child = one; Child-to-Parent = another.

**'MsgBox** "Sub 'CreateFam' has completed." '**Enable for Debugging << ----------------------------------------<<**

End Sub '**End CreateFam() ========================================================================================================**

**Private Function WriteNotes(TextString, First, Last) As String** '**======= Parse text per NoteLenMax max segment length =======** |

'WriteNotes() **extracts text** from within current '**TextString**' passed to it, **between** 'First' and 'Last' **character positions**.

'If Sources & Notes cell text does **NOT** begin with "**NOTE** " or "**SOUR** " tag: VBA code calling WriteNotes() provides "**1 NOTE** ";…

'Else "**1** " is provided by code that calls WriteNotes() while "**NOTE** " or "**SOUR** " is at start of 'TextString' for WriteNotes().

Dim **Partway** As Long, **SegLen** As Long, **sTxt** As String '**Partway**: **end position of** 'TextString' **segment** to be processed next.

s**Txt** = "" 'ExportGed() uses **sGed**; replaced here w/**sTxt**.'**SegLen**: **length** of 'TextString', **modifiable** if/when needed.

**Partway** = **First + NoteLenMax - 1** '**NoteLenMax**: segment **length limit**; " **-1**": 'First' is not zero (0).

While **Partway < Last** 'If current segment length doesn't take us **past** end of TextString, …

**SegLen = NoteLenMax** '**SegLen**: allows shift left to get line breaks between non-space chars.

**NoSpaces**: '"NoSpaces: " is a "Line Label". 'Keep shifting left until get line **break between non-space** characters:

**While Right**(Mid(TextString, **First**, **SegLen + 1**), **1**) = " ": **SegLen = SegLen - 1: Wend** 'Space to right?: shift left **1**.

**While Right**(Mid(TextString, **First**, **SegLen + 0**), **1**) = " ": **SegLen = SegLen - 2: Wend** 'Space at end?: shift left **2**.

**If Right**(Mid(TextString, **First**, **SegLen + 1**), **1**) = " " **Then GoTo NoSpaces** 'Still Space(s)?: Keep shifting.

s**Txt** = s**Txt** + Mid(TextString, First, **SegLen**) & vbCrLf 'This processed segment: begin at current 'First', length = SegLen.

s**Txt** = s**Txt** + "**2 CONC** " '**CONC** (vs. **CONT**) allows importing s/w prgm to word wrap as it chooses.

**First** = **First + SegLen** '(was: First = Partway + 1) 'Position of **first** character of **next** text segment to be processed.

**Partway** = **First** + NoteLenMax - 1 'Position of **last** character of **next** text segment, unless >= Last.

**Wend** 'End of "While Partway < Last" loop.

Dim **sFinal** As String: sFinal = **Trim**(Mid(TextString, First, Last - First + 1)) 'TRIM() **remove**s leading and **trailing space(s)**.

**sTxt = sTxt + sFinal & vbCrLf** '**Add final 'Partway' segment of 'TextString' (parsed or not) to sTxt**.

**'MsgBox "Final sTxt for THIS row for the Individual: " & vbCrLf & vbCrLf & sTxt 'Enable to view each NOTE or SOURce as created.**

**WriteNotes** = s**Txt** 'Return text, parsed if needed, for adding to "sGed" of ExportGed().

End Function '**End WriteNotes()** **==================================================================================================**

**Sub btnBrowse\_Click() '==========================================================================================================**

Dim **tmpfile** As Variant 'Variant: "tmpfile" can accept string **or** numeric. 'Can be false.

tmpfile = Application.**GetSaveAsFilename**("", "**GEDCOM Files (\*.ged)**,\*.ged,All files (\*.\*),\*.\*", 0, "**Select or Enter the GEDCOM file name**")

'[Alternate method w/o filters: .FileDialog(msoFileDialogSaveAs)]

If tmpfile <> False Then 'If no error in obtaining filename, Then …

Call Init 'Init(): Initialize wrks1, wrks2, wrks3, **nIRowsCnt**, **sFile**.

wrks1.Cells(**wrks1row0 - 3, 7**) = **tmpfile** 'GEDCOM file's **Path\Filename** from Browse **into this cell** of wrks**1**.

End If

End Sub **'End btnBrowse\_Click() ==================================================================================================**

**Sub btnExport\_Click() '==========================================================================================================**

Dim **msg** As String, **start** As Variant, **spent** As Variant '(Message), (Start Time), (Spent Time)

**start = Timer 'Set Start Time = Current Time** for reference; Current Time when ended - Start Time = Spent Time.

'v3.0: **Begin CheckBeforeOverwrite.**

Call Init 'v3.0: **Re-read** sFile path/filename **to** **purge any residual**.

Dim **Ans** As Integer 'Ans: Variable for binary response to "**Overwrite?**" [vb**Yes** or vb**No**].

If Dir(sFile) <> "" Then 'Dir(sFile): returns filename. If not null, Then …

**Ans** = MsgBox("**The designated output file '**" & **sFile** & "**' already exists. Overwrite?**", vbQuestion + vbYesNo, "**Overwrite?**")

**If** Ans = vb**No** Then Exit Sub

End If 'v3.0: **End CheckBeforeOverwrite**

**If Not ValidFile()** Then Exit Sub

**If ExportGed()** Then 'If ExportGed() completed successfully (**ExportGED = True**), Then …

msg = "**Successfully exported to this GEDCOM file:** " & vbCrLf & **sFile** & vbCrLf & vbCrLf

msg = msg & "**People-Row Entries:** " & **nIRowsCnt** & " (includes Multi-Marriage entries)" & vbCrLf '**NOT** Total Individuals.

msg = msg & "**Unique Individuals:** " & **nUniqInds** & vbCrLf 'MAXimum / highest **Individual ID** number.

msg = msg & "**Unique Families:** " & **nUniqFams** & vbCrLf 'MAXimum / highest **OwnFamID** number.

msg = msg & "**Family Links:** " & **nFLinkCnt** & vbCrLf & vbCrLf '**Family Link**: Parent-Child = one; Child-Parent = another; etc.

spent = (Timer - start) **\* 1000** 'Convert seconds to **milliseconds**.

spent = IIf(spent > 1000, spent **/ 1000** & " seconds", spent & " milliseconds") 'If >1 sec, report in **seconds**; Else **msec**.

**msg** = msg & "**Processing time:** " & **spent** 'How long it took to process data and export the GEDCOM file.

MsgBox **msg**, vbOKOnly, "**Excel to GEDCOM**" 'Message Box for the above information.

End If

End Sub **'End btnExport\_Click() ==================================================================================================**

**Private Function ValidFile() As Boolean '========================================================================================**

On Error Resume Next 'If can't open sFile, go to next line (If Err …).

Call Init 'Init(): Initialize wrks1, wrks2, wrks3, **sFile**, nIRowsCnt, etc.

Open **sFile** For Output As #1

**If Err > 0** Then 'If **\*.ged** file reported as **invalid** when **attempting overwrite of previous export**: it may be **in use**

ValidFile = False '**by another program**, e.g., GenViewer (GedPad is okay). Close prgm or write to different filename.

MsgBox "**Output File/Path** " & **sFile** & vbCrLf & "**>Folder not found, File in Use, or is Read Only. Correct issue, try again.**"

wrks1.Cells(wrks1row0 - **3**, **7**).Activate '….Cells(…).Activate: Select/go to cell for sFile for inspection.

Else

ValidFile = True 'No problem with path and/or filename provided for sFile.

End If

Close #1

End Function **'End ValidFile() ===================================================================================================**

**'================================================ 'End of Excel2GED VBA code. ================================================**